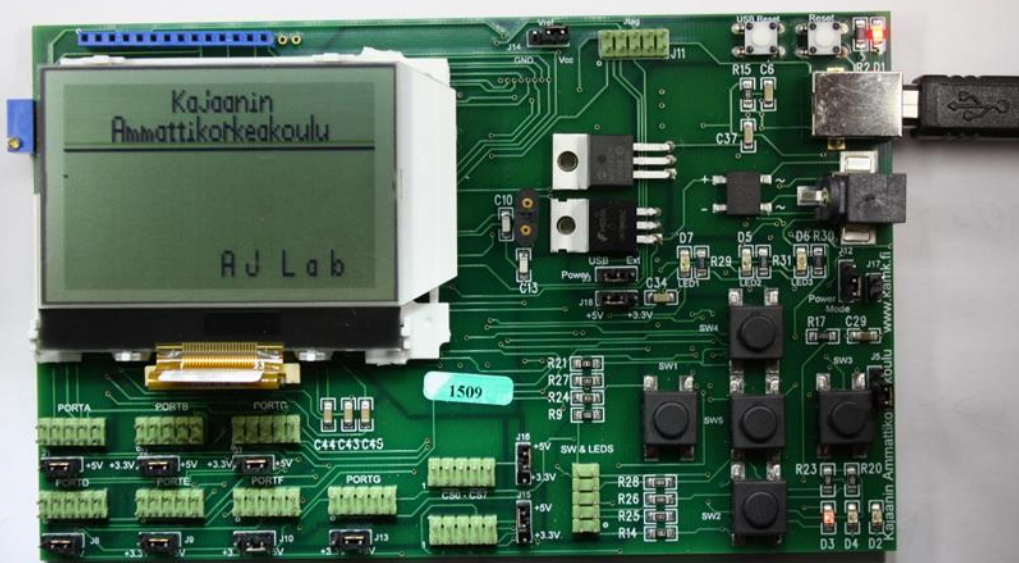


Markku Karppinen

Kehitysalustan suunnittelu - ja toteutus sulautettujen järjestelmien opetuskäyttöön



Insinööri (AMK)

Tieto- ja viestintätekniikka

Kevät 2020

Tiivistelmä

Tekijä(t): Markku Karppinen

Työn nimi: Kehitysalustan suunnittelu ja toteutus sulautettujen järjestelmien opetuskäyttöön

Tutkintonimike: esim. Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: kehitysalustat, laitesuunnittelu, sulautetut järjestelmät, IoT, embedded systems

Kajaanin ammattikorkeakoulun sulautettujen järjestelmien koulutuksessa tuli tarve uudelle mikrokontrolleriopetuksessa käytettävälle kehitysalustalle. Aikaisemmin käytössä ollut 8051-pohjainen ympäristö oli vanhentunut ja sen työkaluohjelmistot eivät toimineet uusissa Windows-työasemissa.

Kehitysalustat ja niiden kehitysympäristöt ovat tärkeässä roolissa tuotekehityksessä ja sulautettujen järjestelmien insinööriopetuksessa. Sulautettujen järjestelmien kehittäminen on hyvin laiteläheistä, ja ohjelmointi perustuu pitkälti mikrokontrollerin sisäisten rekisterien hyödyntämiseen. Teoriaopetuksessa ja harjoitustöissä opiskellaan perusteellisesti mikrokontrollerin tarjoamat rajapinnat. Kehitysalusta mahdollistaa elektroniikan ja ohjelmoinnin opetuksen; koulutuksessa tavoitellaan tilannetta, missä ohjelmoija osaa testata laiterajapinnan kytkennät ja kytkentöjen suunnittelija osaa testata kytkennät ohjelmallisesti.

Tässä opinnäytetyössä on kehitetty kehitysalusta sulautettujen järjestelmien opetukseen. Kehitysalustaa on käytetty Kajaanin ammattikorkeakoulussa tietotekniikan insinööriopetuksessa. Kehitysalustan lisäksi tuotettiin dokumentaatio kortin käyttöönotosta ja käytöstä. Opetuksessa käytettävien järjestelmien ja laitteiden suunnittelussa tulee kiinnittää erityistä huomiota dokumentointiin ja sen selkeyteen.

Kehitysalustan käyttökynnys on erittäin matala. Käyttäjä tarvitsee kehitysalustan lisäksi vain USB-kaapelin, jolla kehitysalusta liitetään työasemaan. Tarvittavat ohjelmistot voidaan ladata ilmaiseksi valmistajan kotisivuilta. Kehitysalusta on vastannut sille asetettuja tarpeita, ja sen puutteita on korjattu uusimmassa kehitysversiona. Liityntöjä on lisätty ja kortin merkintöjä on parannettu. Valittu mikrokontrolleri on kestänyt hyvin aikaa, ja siinä on riittävästi ominaisuuksia opetuskäyttöön. Kehitysalusta on toiminut luotettavasti ja sen sähköisiä ominaisuuksia on testattu EMC-laboratoriossa hyvin tuloksin.

Abstract

Author(s): Karppinen Markku

Title of the Publication: Design and implementation of a Development Platforms for Education use of Embedded Systems

Degree Title: Bachelor of Engineering, ICT Engineering

Keywords: IoT, embedded systems, embedded systems designing, development platforms.

In the education of embedded systems at Kajaani University of Applied Sciences, there was a need for a new development platform for microcontroller training. The previously used 8051-based environment was outdated and the utility software did not work on new Windows workstations.

Development platforms and their development environments play an important role in product development and embedded systems engineering education. The development of embedded systems is very device-oriented, and programming is largely based on the use of a microcontroller's internal registers. The interfaces provided by the microcontroller are carefully studied in theoretical teaching and practical exercises. The development platform enables teaching of electronics and programming.

This thesis has developed a development platform for embedded system teaching. The development platform has been used at Kajaani University of Applied Sciences in computer science education. In addition to the development platform, documentation on the implementation and use of the card was produced. When designing systems and equipment used in teaching, special attention must be paid to documentation and its clarity.

The development platform has a very low usage threshold; in addition to the development platform, the user only needs a USB cable to connect the development platform to the workstation. The necessary software can be downloaded free of charge from the manufacturer's website. The development platform has met its needs and has been included in the latest development version. Interfaces have been added and card entries have been improved. The chosen microcontact roller has been very time consuming and has enough training functions. The development platform has worked reliably, and its electronic properties have been successfully tested in the EMC laboratory.

Sisällys

1	Johdanto	1
2	Sulautetut järjestelmät	2
3	Kehitysalustat	7
3.1	Käyttöjärjestelmäpohjaiset kehitysalustat	10
3.2	Yleiskäyttöiset kehitysalustat	13
3.3	Mikrokontrollerit	14
3.3.1	Rekisteriperustainen ohjelmointi	17
3.3.2	Ohjelmamuisti FLASH	17
3.3.3	Mikrokontrollerin sisäiset toimintalohkot	17
4	Kehitystyökalut	19
4.1	Elektroniikkasuunnittelu	19
4.1.1	Simulaatiot	20
4.1.2	Piirikaavio	21
4.1.3	Piirikortin suunnittelu	22
4.2	Ohjelmistokehitys	21
4.2.1	Ohjelma-editori	21
4.2.2	Kääntäjä	22
4.2.3	Ohjelman lataus ja virheenkorjaustyökalut	25
4.3	Testaus	29
4.3.1	Ohjelmistotestaus	30
4.3.2	Elektroniikkatestaus	31
4.3.3	Hyväksymistestaus	31
4.3.4	Tuotantotestaus	32
5	Kehitysalustan toteuttaminen	33
5.1	ATMega128	41
5.2	Suunnitteluprosessi	42
5.2.1	Piirikortin toteutus	44
5.2.2	Komponenttien valinta ja hankinta	44
5.3	Tuotanto ja testaus	45
5.3.1	Piirikorttien ladonta	45

5.3.2	Testaus	46
5.3.3	Toiminnallinen testaus	46
5.3.4	Ympäristötestaus.....	48
5.3.5	Dokumentointi	48
6	Pohdinta	49
7	Havaitut kehityskohteet	52
8	Yhteenveto	53
	Lähteet	54

Symboliluettelo

1-wire, yhden johtimen sarjamuotoinen tiedonsiirtoväylä.

3-D, 3-ulotteinen mekaaninen mallintaminen.

8051-mikrokontrolleri, Intelin ensimmäinen mikrokontrolleriarkkitehtuuri.

ADC, Analog Digital Conversion, analoginen signaali muutetaan digitaaliseen muotoon.

ARM, Advanced RISC Machines, tehokkaaseen laskentaan kykenevä mikrokontrolleriarkkitehtuuri.

ASIC, Application Specific Circuit, erikoistehtävään valmistettu mikropiiri.

BOM, Bill Of Material, piirilevyohjelmiston tuottama komponenttiluettelo.

CAD, Computer Aid Design, tietokoneperustainen mekaaninen piirtäminen.

CAN, Controller Area Network, teollisuudessa ja ajoneuvoissa käytetty tiedonsiirtoväylä.

CPU, Central Processing Unit, tietokoneen laskentayksikkö.

EMC, sähkömagneettinen häiriö.

ESD. Sähköstaattinen purkaus, suojaudutaan maadoituksin.

FPGA, Field Programmable Gate Array, ohjelmoitava logiikkapiiri.

GCC, Gnu Compiler Collection, vapaassa jakelussa oleva kääntäjäkokoelma.

GPIO, General Purpose Input Output, yleiskäyttöinen kaksisuuntainen mikrokontrollerin liityntä.

HART, High Addressable Remote Transducer, teollisuudessa käytetty tiedonsiirtoväylä.

HDMI, High Definition Interface, laadukkaan kuvan ja äänen liityntä.

I2C, (TWI, 2-wire), laitteen sisäinen 2-johtiminen sarjamuotoinen tiedonsiirtoväylä.

IDE, Integrated Development Environment, ohjelmistokehitysympäristö.

IoT, Internet of Things, esineiden internet, ”Anturista pilveen”.

Jtag, Join Test Action Group, mikrokontrollerin testaus- ja ohjelmointiliityntä.

LED, Light Emitted Diodi, valodiodi.

LIN, Local Interconnect Network, ajoneuvoissa käytetty tiedonsiirtoväylä.

LINUX, avoimeen lähdekoodiin perustuva käyttöjärjestelmä.

PC, Personal Computer, henkilökohtainen työasema.

RISK, Reduced Instruction Set Computer, suoritinarkkitehtuuri.

RS232, laitteiden välinen sarjaliikennestandardi.

Spi, Serial Peripheral interface, laitteen sisäinen 4-johtiminen sarjaliikenneväylä.

TTL, Transistor – Transistor Logic, logiikkajännitetasojärjestelmä.

RTOS, Real Time Operation Systems, reaaliaikainen käyttöjärjestelmä.

SD-card, Secure Digital, kannettaviin laitteisiin kehitetty muistikortti.

USB, Universal Serial port, laitteiden välinen sarjaliikennerajapinta.

VHDL, Very Hardware Description Language, logiikkakytkentöjen kuvauskieli.

WEB, maailmanlaajuinen tietoverkko.

WiFi, Langaton tiedonsiirtojärjestelmä, joka perustuu IEEE 802.11 -standardiin.

1 Johdanto

Sulautetut järjestelmät ovat syrjäyttäneet perinteistä elektroniikkaa. Osa toiminnoista, joita aikaisemmin toteutettiin erillisillä logiikkapiireillä tai analogiaelektroniikan kytkennöillä, toteutetaan ohjelmallisesti mikrokontrollerissa. Sulautettu järjestelmä on laitekokonaisuus, joka koostuu elektroniikka- ja ohjelmisto-osasista. Nykyään käytetään usein nimitystä IoT (Internet Of Things), esineiden internet. Laitteen sydän on mikrokontrolleri, joka ohjaa toimintoja sisäisen ohjelmiston mukaan.

Ohjelmistokehitys on hyvin laiteläheistä, koska käyttöjärjestelmää ei yleensä ole. Ohjelmoijan tulee ymmärtää elektroniikkaa ja elektroniikkasuunnittelijan ohjelmointia ainakin sen verran, että kykenee testaamaan omat suunnitelmansa.

Sulautetut järjestelmät ovat merkittävässä osassa myös Kajaanin ammattikorkeakoulun Älykkäät järjestelmät -suuntautumisvaihtoehdon opetuksessa. Kehitettäessä sulautettujen järjestelmien opetusympäristöä havaittiin tarve uudelle kehitysalustalle. Aikaisemmin käytetty 8051-pohjainen järjestelmä oli vanhentunut.

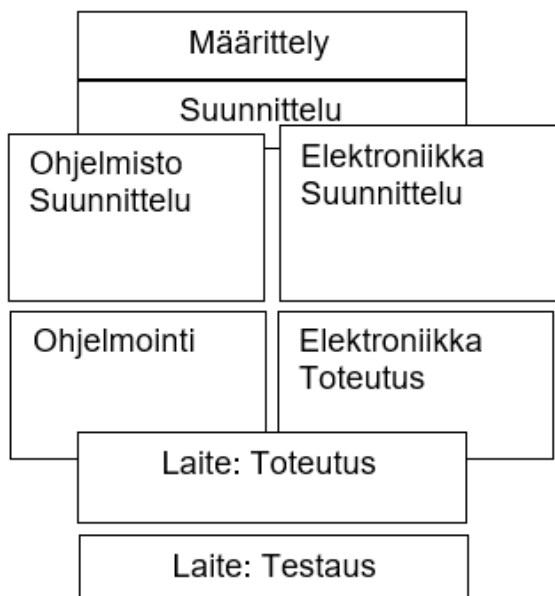
Lähtökohta oli kehitysympäristö, jonka käyttö ja käyttöönotto olisi mahdollisimman helppoa. Kehitysalustan käyttö ei vaadi lisenssiperustaisten ohjelmien käyttöä, vaan tarvittavat ohjelmistot saadaan ladattua valmistajan kotisivuilta. Käyttäjä tarvitsee kehitysalustan lisäksi USB-kaapelin, jolla kortti liitetään työasemaan.

Suunnitteluvaiheessa vertailtiin kaupallisia kehitysalustoja sekä muissa korkeakouluissa käytössä olevia kehitysalustoja. Lisäksi opetuksen tarpeista keskusteltiin oman koulun opettajien ja Kajaanin ammattikorkeakoulun saksalaisen yhteistyökorkeakoulun [13], HS-Heilbronn HHN kollegojen kanssa. Vertausten tuloksena päädyttiin toteuttamaan opetuksen tarpeet täyttävä kehitysalusta itse.

2 Sulautetut järjestelmät

Olemme tekemisissä sulautettujen laitteiden kanssa päivittäin. Esimerkiksi ajoneuvoissa ja kodinkoneessa / laitteessa on paljon sulautettua laitetekniikkaa. Kodinkoneet sisältävät useita antureita, jotka mittaavat jotakin fyysistä suuretta. Laitteen sisällä on mikrokontrolleri, joka tekee antureiden antamien mittaustulosten perusteella ohjelmoituja toimintoja ja ohjauksia.

Mikrokontrolleriympäristön kehitysprosessi jakautuu projektin suunnittelu- ja määrittelyvaiheen jälkeen elektroniikka- ja ohjelmistosuunnitteluosiin. Parhaassa tapauksessa elektroniikka- ja ohjelmistosuunnittelu etenevät rinnakkain (kuva 1). Ohjelmistosuunnittelu voidaan aloittaa valitun mikrokontrolleriympäristön kehitysalustoilla jo ennen varsinaisen elektroniikkaosan valmistamista. Kehitysalustat ovat tuotekehitysvaiheen työkaluja, ja niillä on myös keskeinen rooli laitelaheisen ohjelmoinnin opetuksessa.

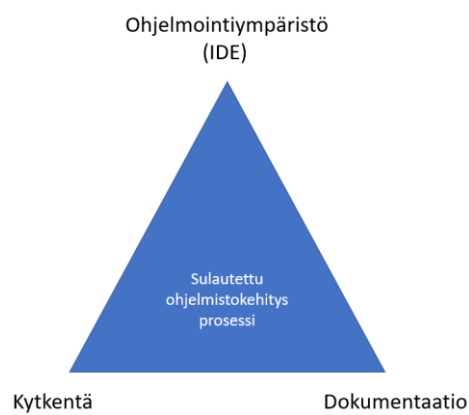


Kuva 1. Sulautetun laitteen suunnitteluprosessi.

Järjestelmän suunnitteluprosessin alkuvaiheessa tehdään laitteistonmäärittely ja suunnittelu. Tässä vaiheessa otetaan kantaa käytettävään laitteistoon ja sen suorituskykyyn. Sulautetun järjestelmän suunnittelijan tarvitsemat dokumentaatiot kerätään useista eri lähteistä. Sovellettavat standardit riippuvat laitteiston käyttöympäristöstä [2] ja valituista tiedonsiirtojärjestelmistä.

Komponenttivalmistajat tarjoavat dokumentaation lisäksi työkalut ohjelmointiin ja tuotekehitykseen (kuva 2).

- Dokumentaation, datalehdet, esimerkkisovellukset.
- Valmistajan piirikohtaiset kehitysalustat.
- Ohjelmointiympäristö (IDE).



Kuva 2. Ohjelmistokehitys

Opetuskäytössä kehitysalustat mahdollistavat erilaisten mikrokontrolleri rajapintojen opiskelun. Kehitysalustoja ja niiden ohjelmistokehitysympäristöjä käytettäessä päästään nopeasti ja helposti tutustumaan laiteläheiseen kehitysympäristöön esimerkkisovellusten avulla. Sulautettujen järjestelmien ohjelmistokehitysympäristöä kutsutaan yleisesti englanninkielisellä termillä IDE (Integrated Development Environment).

Komponenttivalmistajat markkinoivat tuotteitaan julkaisemalla toimivia ja dokumentoituja toteutuksia tuotteista. Näin loppukäyttäjä pystyy helposti testaamaan tuotteen sopivuuden omassa projektissa. Helpoin tapa aloittaa tutustuminen sulautettuihin järjestelmiin on hankkia kokeilukortti. Tuotekehityksessä käytettävistä elektroniikka-alustoista käytetään erilaisia nimityksiä:

- Kokeilukortti (starter board)
- Demokortti (demoboard)
- Kehitysalusta (evaluation board).

Kokeilukortti on valmis kokonaisuus, joka toimii itsenäisenä laitteena. Usein se on liitettävissä tietokoneeseen tai mobiilialustaan valmiilla sovelluksella. Kuvassa 3 on Silicon Labsin kokeilukortti, joka mittaa mm. ympäristön lämpötilan, ilmankosteuden ja pulssin sormenpäältä [3]. Laite lähettää mittausdatan langatonta tiedonsiirtoa käyttäen mobiililaitteelle, kännykkään tai tablet-tietokoneelle. Valmis sovellus on ladattavissa valmistajan kotisivuilta (kuva 3).

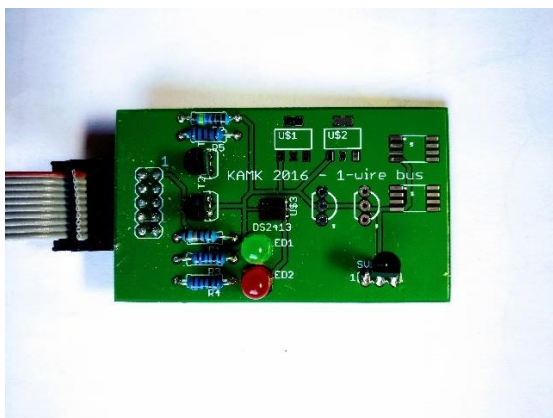


Kuva 3. Silicon Labs -kokeilukortti



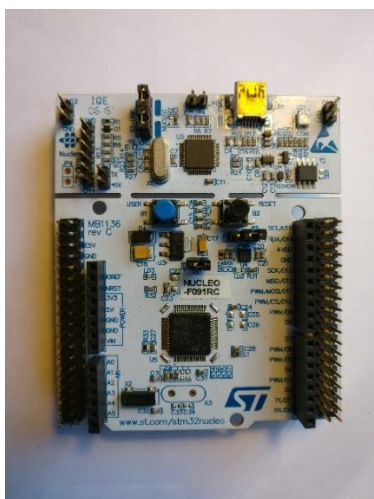
Kuva 4. Pulssinmittauskäyttöliittymä

Demokortti on yhden spesifisen ilmiön tutkimiseen ja havainnointiin tarkoitettu ympäristö. Demokorteilla valmistajat esittelevät uusia tuotteita. Ne liitetään osaksi suurempaa järjestelmää, esim. tiedonsiirtoväylän toimintaa. Kuvassa 4 on opetuskäyttöön tehty 1-wire-väylän demokortti. Kortilla on toteutettu väyläsovitin ja kortilla on antureita, muistipiirejä ja IO-piiri.

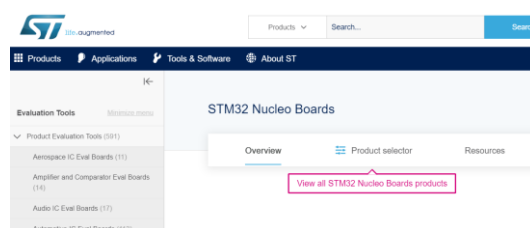


Kuva 5. 1-wire-väylän demokortti

Kehitysalustat sisältävät mikrokontrolleripohjaisen piirikorttitason toteutuksen (kuva 6) ja ohjelmistotyökalut ja dokumentoinnin esimerkkiratkaisuin (kuva 7). Loppukäyttäjä löytää ohjeita ja neuvoja piirin valmistajan kotisivuilta tai useilta käyttäjäryhmäsivustoilta [4].



Kuva 6. ST Nucleo -kehitysalusta, hinta noin 10 euroa.



Kuva 7. Kehitysalustan kotisivut

Kehitysalustat ovat yleensä edullisia, halvimmat noin kymmenen euroa. Edullinen hintataso mahdollistaa valmistajan tuotteiden laajan käytön ja kulutuksen. Arvokkaimmat ohjelmoitavan logiikan VHDL-pohjaiset kehitysalustat maksavat kymmeniä tuhansia euroja.

Mikrokontrollerivalmistajilla on omat ohjelmistokehitysympäristönsä, jotka tyypillisesti sisältävät seuraavassa esitellyt ominaisuudet;

Editori, tekstinkäsittely, ohjelman kirjoittamiseen ja editointiin.

- Kääntäjä, kääntää ohjelman konekieliseen muotoon.
- Ohjelman lataus, siirtää konekielisen tiedoston mikrokontrollerin FLASH-muistiin
- Debuggeri, ohjelman virheenkorjaustyökalu.

Kääntäjät pohjautuvat usein vapaan lähdekoodin ohjelmistoihin, esim. Gnu C Compiler (GCC), ja ne ovat aina mikrokontrollerikohtaisia. Markkinoilla on myös kolmannen osapuolen ohjelmistotuotteita, joista ilmainen versio on käytettävissä vain lyhyen määräajan. Ohjelmiston lataus- ja testaustyökalut vaativat kontrollerin Jtag-liittymän. Jtag mahdollistaa ohjelmiston ajon aikaisen seuraamisen - debuggauksen. Markkinoilla on myös yleiskäyttöisiä kehitysalustoja, joihin voidaan vaihtaa valmistajan eri tyyppin kontrollereita ja rakentaa omia kytkentöjä alustan ympärille.

3 Kehitysalustat

Kehitysalusta on piirikortille toteutettu esimerkkikytkentä, jota käytetään tuotekehityksessä ja opetuksessa. Mikrokontrollerin valmistaja on toteuttanut toimivan kokonaisuuden, jossa on tarvittavat:

- liitännät
- virtalähde
- dokumentaatio
- ohjelmointityökalut.

Kehitysalusta on aina tuotekehityksen työkalu, eikä se koskaan päädy lopulliseen tuotteeseen. Lopulliseen tuotteeseen suunnittelija usein ”kopioi” osia kehitysalustan kytkennöistä. Kehitysalusta voi perustua prosessori- tai mikrokontrolleripohjaiseen järjestelmään. Prosessoripohjaiset kehitysalustat toimivat käyttöjärjestelmän päällä, yleisin käyttöjärjestelmä on Linux. Käyttöjärjestelmästä on useita käyttö- ja ympäristöversioita (distro). Kehitysalustalle testatut käyttöjärjestelmät ovat ladattavissa valmistajan kotisivuilta. Kuvassa 8 on Raspberry Pi -kehitysalusta. Alustaan on integroitu hyvät liitännät oheislaitteille.



Kuva 8. Raspberry Pi 3 -kehitysalusta

Raspberry Pi 3 on Farnell 14 Org Ltd:n kehittämä kehitysalusta, ja se on hyvin yleisesti kehittäjien käytössä. Käyttöjärjestelmä ladataan käynnistyksen yhteydessä SD-kortilta. Käyttöjärjestelmä-pohjaiset kehitysalustat vaativat tehokkaan suorittimen, ja ne ovat usein moniydinsuorittimia. Raspberryn kotisivuilta on ladattavissa useita eri käyttökohteisiin räätälöityjä käyttöjärjestelmän asennuspaketteja, esimerkiksi

- Noobs (New Out Of the Box Software)
- Raspbian.

Usein kehitysalustan kytkentöjä tai sen osia ”kopioidaan” lopulliseen tuotteeseen, mikä edellyttää hyvää dokumentaatiota.

Laitevalmistajat tarjoavat kehitysalustoja eri käyttötarkoituksiin:

- Digitaalinen signaalinkäsittely (DSP)
- Tiedonsiirto (Scurity)
- Moottorinohjaus (motor control)
- Tietoturva (Gryptography embedd security)
- Ohjelmoitava logiikka (FPGA ASIC)
- Väylät (CAN, LIN, HART)
- Vähävirtaiset laitteet (Energia Harvesting)

Sulautetuissa järjestelmissä kehitysalustan valinta tehdään projektin vaatimusten perusteella. Valintaan vaikuttavat mm. mikrokontrollerin laskentateho, I/O-määrä ja ulkoiset liitännät. Mikrokontrolleripohjaiset kehitysalustat vaativat laiteläheistä tapaa lähestyä ongelmaa. Ohjelmointi tehdään mikrokontrollerin sisäisiä rekisterejä hyödyntäen. Rekisteripohjainen ohjelmointi vaatii perehtymistä kontrollin datalehteen. Datalehdessä kuvataan kontrollerin toiminta ja annetaan esimerkki koodin C- ja Assembly-ohjelmointikielellä (kuva 9).

```
/*This is an AVR C code*/
```

```
int main()
{
    int i = 0;
    for (i = 0; i < n; i++)
    {
        USARTWriteChar(string[i]);
    }
}
```

```
; This is an AVR assembly code
```

```
.include "m8515def.inc"
.org $00

        ADD    R16, R17    ; Add value in R16 to value in R17
        DEC    R17        ; Minus 1 from the value contained in R17
        MOV    R18, R16    ; Copy the value in R16 to R18
END:     JMP     END       ; Jump to the Label END
```

Kuva 9. C- ja Assembly-ohjelmointikieli

Yleiskäyttöinen kehitysalusta mahdollistaa usean mikrokontrollerin käyttämisen samassa alustassa, erityyppiset piirit liitetään alustaan sovituskorteilla. Sovitinkortit tulevat kehitysalustan mukana. STK600-kehitysalustan (kuva 10) levyllä on kaikki I/O- liitynnät tuotu liittimille, mistä niitä voidaan kytkeä omaan ulkoiseen kytkentään. Alustalla on valmiita toimintoja, jotka voidaan liittää mikrokontrollerin liityntöihin kaapeleilla.

- 8 kpl kytkimiä ja LEDejä
- RS232-, CAN- ja LIN-liitynnät.
- Ohjelmointilaiteliitynnät: Jtag, Isp
- DataFlash
- Virtalähde.



Kuva 10. Microchip AVRSTK600 -kehitysalusta

3.1 Käyttöjärjestelmäpohjaiset kehitysalustat

Tehokkailla ARM-pohjaisilla kehitysalustoilla käytetään usein Linux-pohjaisia käyttöjärjestelmiä, jotka tukevat usein reaaliaikaohjelmistokehitystä (RTOS). Käytetyt ARM-perustaiset mikrokontrollerit ovat 32- tai 64-bittisiä, niiden laskentatehoa voidaan verrata älypuhelinien tehoon. Käyttöjärjestelmän edut tulevat esille suurissa projekteissa. Käyttäjän ei tarvitse ohjelmoida kaikkia toimintoja itse, vaan käyttöjärjestelmä tarjoaa valmiin rajapinnan esim. ulkoisen näytön käsitteilyyn.

Raspberry PI Foundationin kehittämä kehitysalusta on suunniteltu tukemaan tietotekniikan opetusta. Raspberry on saanut ison jalansijan harrastajien ja ammattilaisten keskuudessa, ja siitä on julkistettu useita versiota sen kymmenen vuoden historian aikana.

Tekniset tiedot, Raspberry Pi 3 (kuva 8)

- 1,2 GHz moniydin ARM-64Bit-prosessori
- Näytönohjain Brodcom
- Liitännät: HDMI, 4xUSB, 3,5mm Audio, Ethernet, Wifi, Bluetooth
- GPIO 40 pins
- Hinta n. 50 €

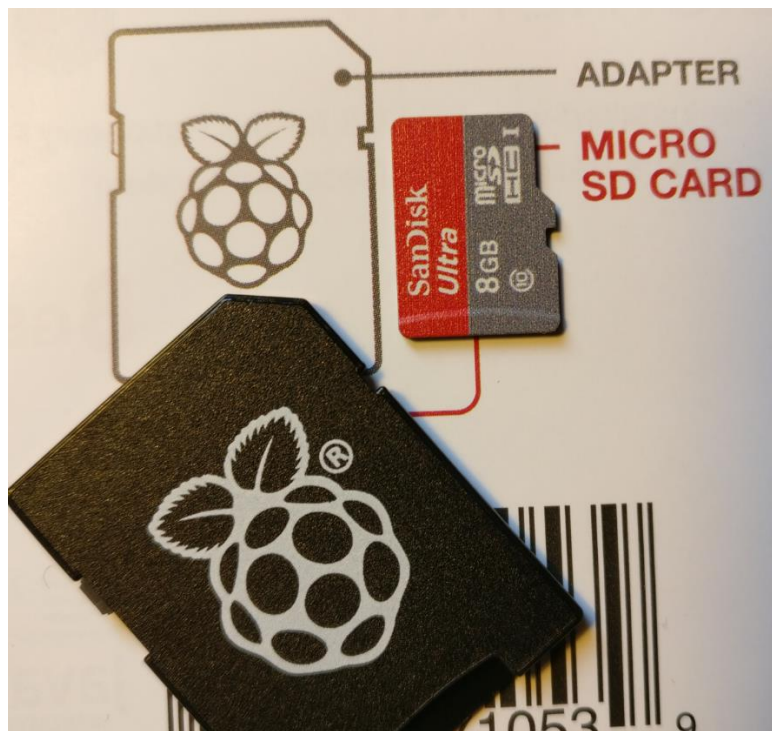
Kehitysalustoille on useita Linux-pohjaisia käyttöjärjestelmäpaketteja, ja jopa sovelluskohtaisia versioita on saatavilla ilmaiseksi, esim. kodinviihdekeskus (KODI), opetus tai sääasema. Käyttöjärjestelmiä sulautettuun laitekehitykseen on vapaasti tarjolla. Käyttöjärjestelmän asennuspaketin voi ladata kehitysalustan valmistajan kotisivuilta [5].

- FreeRTOS (c-kielinen)
- OpenWRT
- GNU Arm /Linux
- Noobs, Rasbian

- Windows 10 IoT Core
- Chromium OS

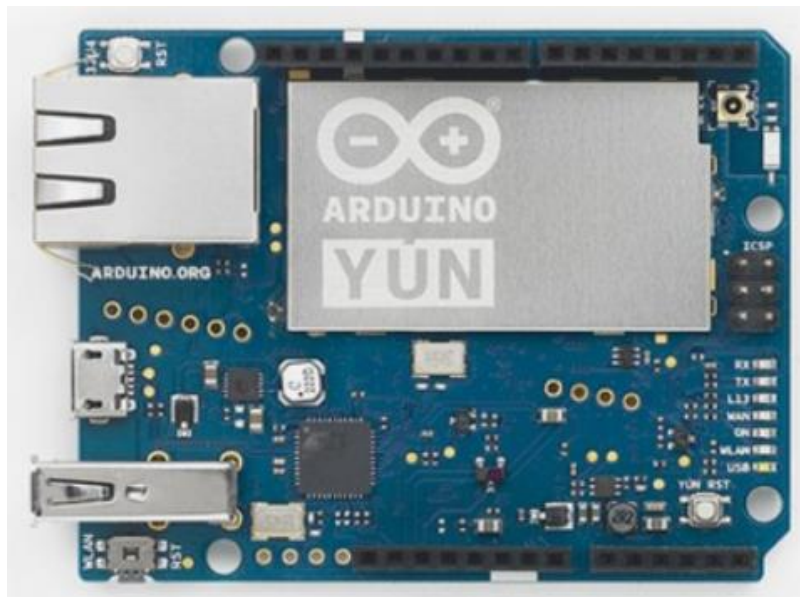
FreeRTOS on vapaasti käytettävä reaaliaikakäyttöjärjestelmä sulautettuun laiteympäristöön. Ohjelmistokehitys tapahtuu c- tai c++-kielellä. FreeRTOS-käyttöjärjestelmä on käytettävissä Microchip- (Atmel), NXT- ja STMicro-valmistajien komponenteilla. OpenWRT on kehitetty lähinnä langattoman (Wifi) tiedonsiirron tarpeisiin; Arduino Yun -kehitysalustan Linux-osio käyttää OpenWRT-käyttöjärjestelmää (kuva 10). Gnu Arm -työkaluohjelmistot on kehitetty ARM Cortex -piirisarjan ohjelmistokehitykseen. Noobs- ja Rasbian Linux -versiot on suunniteltu käytettäväksi Raspberry-kehitysalustan kanssa. Valmiit levykuvat ovat ladattavissa valmistajan kotisivuilta (www.raspberrypi.org). Kotisivuilta löytyvät myös kolmannen osapuolen tarjoamat käyttöjärjestelmän asennuspaketit, esim Ubuntu Mate, Windows 10 IoT Core ja Weather station. Chromium OS on Googlen tarjoama vapaasti käytettävä käyttöjärjestelmä 32-bit Arm-, 32- ja 64-bit Inter/Amd-prosessoreille.

Käyttöjärjestelmät ladataan valmistajan nettisivuilta, ja asennuspaketti puretaan SD-muistikortille. Kortti siirretään kehitysalustaan, ja ensimmäisen käynnistyksen yhteydessä käyttöjärjestelmä konfiguroidaan SD-kortille (kuva 11). Kehitysalustan käynnistyksen yhteydessä käyttöjärjestelmä ladataan kortin muistiin ja otetaan käyttöön.



Kuva 11. 8 GB MicroSD-kortti ja SD-sovitin

Sovelluskehittäjät ja harrastajat julkaisevat projektejaan käyttäjäryhmien omilla verkkosivuilla ja keskusteluryhmissä. Raspberry ja Arduino ovat yleisimmät käytetyt kehitysalustat; vähemmän käytettyjä ovat esim. Beagle board ja CubieBoard.



Kuva 12. Arduino Yun -kehitysalusta

Arduino Yun poikkeaa muista kehitysalustoista (kuva 12). Samalle alustalle on yhdistetty 8 bit mikrokontrolleri ja 32 bit mikroprosessori, jossa on käytössä Linux-pohjainen käyttöjärjestelmä. 8 bitin ja 32 bitin lohkojen välillä on sarjaliikennerajapinta. Käyttöjärjestelmän alainen osa huolehtii mm. WiFi-tiedonsiirrosta. Tähän on myös mahdollisuus toteuttaa web-serveri, jolla on mahdollisuus toteuttaa käyttöliittymä.

3.2 Yleiskäyttöiset kehitysalustat

Mikrokontrolleripohjainen ohjelmistokehitys on laiteläheistä ohjelmointia, jos mikä. Ohjelmoija lukee mikrokontrollerin valmistajan datakirjaa kuin ”piru Raamattua” ja toteuttaa C-kielisen ohjelmakoodin, joka pohjautuu suurelta osin rekisterien käsittelyyn. Ohjelmointikieli on yleensä C- tai Assembler, joissain tapauksissa näiden sekoitusta.

Mikrokontrolleri on kehitysalustan tärkein komponentti. Piiriin on integroitu paljon erilaisia oheistoimintoja, esim.

- ohjelma- ja käyttömuisti
- tiedonsiirto SPI, Usart, Twi...
- AD-muunnin
- Keskeytysten käsittely.

Mikrokontrollerien valmistajat esittelevät tuotteiden käyttöä kehitysalustojen avulla. Valmistajan dokumentaatiot ovat internetissä kotisivuilla. Sieltä voi tilata yksittäisiä komponenttinäytteitä veloitusetta. Valmistaja tarjoaa piirin valintaan valintataulukot, joiden perusteella voidaan siirtyä valitun piirin sivuille tarkkoihin spesifikaatioihin:

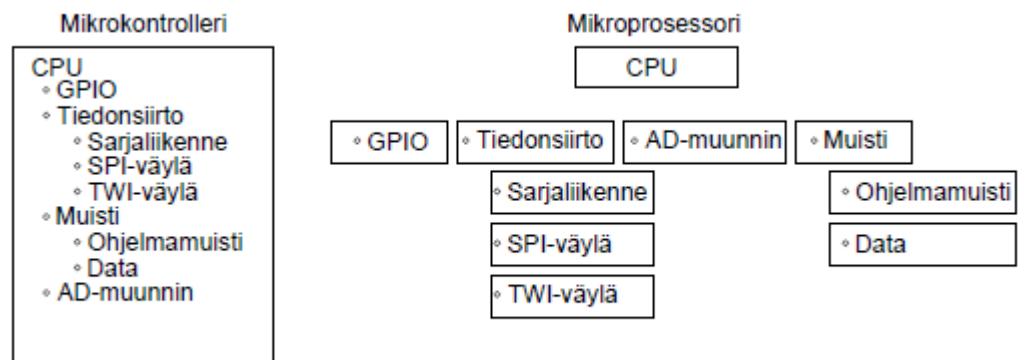
- Dokumentaatio,
 - datalehdet, lyhyt ja täydellinen versio
 - esimerkkisovellukset (Application notes)
 - käyttöönotto-ohjeistus (Quick Start guides)
 - ohjelmointioppaat.

- Hankintadokumentaatiot
 - hinta yksittäin ja suurissa erissä
 - näytekappaleiden tilaus
- Tuotekehitystyökalut
 - kehitysalustat
 - ohjelmointilaitteet
- Valintataulukot
 - samankaltaiset tuotteet

Kehitysalustan lisäksi tulee kiinnittää huomiota ohjelmointiympäristöön (IDE). Usein käytetään valmistajan tarjoamia ohjelmistoja, mutta on olemassa kolmannen osapuolen työkaluohjelmistoja. Ne sisältävät laajempia ominaisuuksia kuin valmistajan tuotteet mutta ovat hankintahinnaltaan kalliita.

3.3 Mikrokontrollerit

Mikrokontrolleri ja mikroprosessori eroavat siten, että mikrokontrollerissa samaan piiriin on integroitu useita sisäisiä toimintalohkoja (Peripheral features) (kuva 13). Mikroprosessoriympäristössä tarvittavat erilliset oheislaitteet ja muistit yhdistetään dataväylän avulla. Mikroprosessoria käytetään esim. tietokoneissa ja palvelinsovelluksissa.



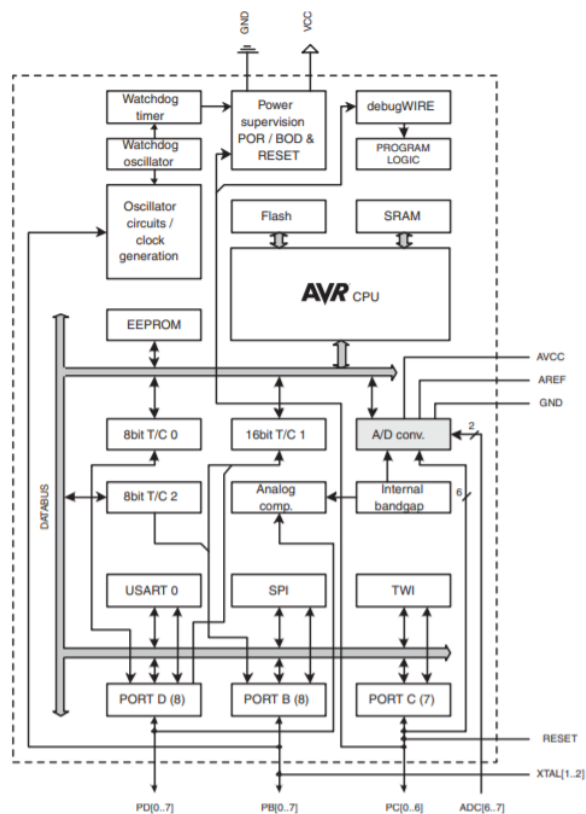
Kuva 13. Mikrokontrolleri vs. prosessori

Mikrokontrollerit vaativat vain vähän ulkoisia komponentteja toimiakseen. Vähintään tarvitaan:

- virtalähde (Power)
- kello (XTAL)
- RESET.

Virtalähteellä tuotetaan piirin tarvitsema käyttöjännite. Se voidaan toteuttaa käyttäen akkua tai muuta ulkoista virtalähdettä. Kellosignaalia varten rakennetaan ulkoinen oskillaattori kiteellä ja kondensaattoreilla. Joissakin mikrokontrollereissa on mahdollisuus käyttää sisäistä oskillaattoria, jonka kellotaajuutta voidaan muokata. RESET-signaalin käsittelyyn on olemassa valmiita komponentteja, ja usein RESET-signaali varmistetaan ylös- tai alasvetovastuskytkennällä riippuen linjan aktivointitilasta.

Mikrokontrollerissa ytimen (CPU) ympärillä on useita sisäisiä toimintalohkoja (Peripheral features); kaikki ovat yhden komponentin sisällä. Kuvassa 14 on ATmega 48:n sisäinen lohkokaavio. Kyseinen mikrokontrolleri on piirisarjan pienimmästä päästä.



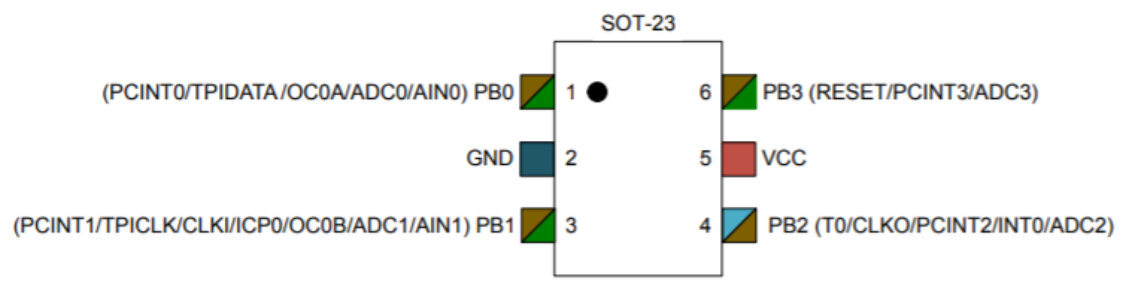
Kuva 14. ATmega 48 -mikrokontrollerin lohkokaavio [6]

Mikrokontrollerit ovat syrjäyttäneet perinteistä erilliskomponenteilla tehtävää elektroniikkaa. Komponenttivalmistajilla on useita piirisarjoja (family) eri käyttökohteiden mukaan. Mikrokontrollerit ovat suhteellisen edullisia suurten valmistussarjojen takia. Taulukkoon 1 on koottu Microchip-tuoteperheiden komponenttien yksittäisiä hintoja ja muita perusominaisuuksia.

Taulukko 1. Microchip-piirisarjojen ominaisuuksia [7]

Malli	Hinta yksittäin hankittuna	liityntäpisteiden määrä	Ohjelmamuisti (Flash)
AVR Tiny	n. 0,20 €	6 - 20	0,5 – 16 KB
AVR Mega	n. 2,00 €	28 - 100	8 – 256 KB
AVR X Mega	n. 5,00 €	32 - 100	16 - 384 KB
SAM (ARM 32bit)	n. 12,00 €	32 - 144	0.256 – 2MB

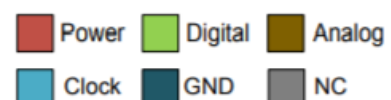
AVR Tiny on valmistajan pienin mikrokontrolleripiiriperhe. Taulukossa 2 on piirisarjan pienimmän mikrokontrollerin tarkempia suoritusarvoja sekä käyttöympäristöparametrejä. Pienimmillään mikrokontrolleri on pakattu 2 x 2 mm kooltaan olevaan pintaliitoskoteloon. Mikrokontrollerin yhdellä kytkentäpisteellä on useita eri toimintoja (kuva 14), jotka otetaan käyttöön ohjelmallisesti alustuksen yhteydessä.



Kuva 14. AVR Tiny4 -mikrokontrolleri [6]

Taulukko 2. Tiny4 ominaisuudet [6]

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	0.5
CPU Speed (MIPS)	12
RAM Bytes	32
Capture/Compare/PWM Peripherals	1 Input Capture, 1 CCP, 2PWM
Timers	1 x 16-bit
Comparators	1
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	6



3.3.1 Rekisteriperustainen ohjelmointi

Mikrokontrollerin sisäisten toimintalohkojen ohjelmallinen käyttöönotto ja käyttö perustuu pitkälti sisäisten rekisterien käyttöön ja käsittelyyn. Ohjelmoijan käytössä on kymmeniä rekisterejä, ja niiden käyttö ja hallinta vaatii datalehtien jatkuvaa seuraamista. Esimerkiksi ulkoisten liityntöjen (GPIO) käyttöönotto ja käyttö vaatii muutamien rekisterien käsittelyä [1.]

- Suuntarekisteri, DDRx, kertoo portin yksittäisten liityntäpisteiden suunnan, input "1" / output "0".
- Tulo (input) PINx-rekisteri. Luetaan yksittäisen tulon tila.
- Lähtö (output) PORTx-rekisteri. Ohjataan lähdön tila "1" tai "0"

Alla esimerkkinä (8-bit) A-portin käsittelyyn liittyvät rekisterit

```
DDRA = 0b11100000; //alustus, suuntarekisterit: 3 ylintä ulostulo ja 5 alinta sisään
```

```
Pin3 = (PINA & (PA0)) //yksittäisen bitin luenta
```

```
PORTA |= (1<<PA5) | (1<<PA6) | (1<<PA7); //Ohjataan ulostulot tilaan "1".
```


3.3.2 Ohjelmamuisti FLASH

Mikrokontrollerin ohjelma talletetaan piirin sisäiseen ohjelmamuistiin (Flash), josta se ladataan käynnistytyn tai resetoinnin yhteydessä. Flash-muistin käyttöikä on tyypillisesti 10 000 kirjoituskertaa ja tieto säilyy muistissa lähes ikuisesti (20 vuotta). Flash-muisti on jaettu kahteen osaan: ohjelmamuisti ja käynnistysasetukset (boot loader). Taulukosta 2 (TINY4:n tekniset tiedot) selviää, että piirissä on 0,5 Kb:n ohjelmamuisti, mikä ei ole kovin paljon. Ohjelmamuistin koko on usein rajoittava tekijä ohjelmistosuunnittelun kannalta; sulautetuissa laitteissa ei yleensä käytetä kiintolevytyyppisiä massamuisteja. Ohjelmamuistia voidaan tarvittaessa laajentaa ulkoisilla muistipiiriratkaisuilla.

3.3.3 Mikrokontrollerin sisäiset toimintalohkot

Mikrokontrollerit sisältävät useisiin eri käyttötarkoituksiin sovellettavia sisäisiä toimintalohkoja. Kuvassa 15 on Microchipin AVR-kontrollereiden valintataulukko, jonka avulla piirin valintaa voidaan alustavasti hahmotella. Tarkemmat tiedot löytyvät mikrokontrollerin datalehddestä, ja ne ovat useita satoja sivuja pitkiä dokumentteja.

8-bit AVR® Microcontrollers Peripheral Integration

Quick Reference Guide

Product Family	Pin Count	Program Flash Memory (KB)	SRAM (KB)	Peripheral Function Focus																										
				Intelligent Analog							Waveform Control		Timing and Measurements		Logic, Crypto and Math		Safety and Monitoring		Communications											
				ADC (# of bits)	ADC (# of channels)	Comparators	ADC Gain Stage	DAC (# of bits)	Temperature Sensor	Internal Voltage Reference	8-bit PWM	16-bit PWM	Quadrature Decoder	Waveform Extension	Real-Time Counter	8-bit Timer/Counter	12-bit Timer/Counter	16-bit Timer/Counter	CCL	MULT	Crypto (AES/DES)	CRC	POR	BOD	WDT	UART	USART	USB	I ² C	SPI
ATtiny4/5/9/10	6	0.5-1	0.032	10 ³	4 ⁽¹⁾	✓						2										✓	✓	✓						
ATtiny102/104	8/14	1	0.032	10	5/8	✓				✓		2					2					✓	✓	✓		1				
ATtiny13A	8-20	1	0.064	10	4	✓						2										✓	✓	✓						
ATtiny20/40	12-20	2/4	0.128/0.256	10	8/12	✓			✓			2	2		1	1						✓	✓	✓				1	1	
ATtiny24A/44A/84A	14-20	2-8	Up to 0.512	10	8	✓			✓			2	2		1	1						✓	✓	✓				1	1	
ATtiny25(V)/45(V)/85(V)	8-20	2-8	Up to 0.512	10	4	✓			✓			4			2							✓	✓	✓					1	
ATtiny48/88	28-32	4/8	Up to 0.512	10	8	✓			✓			1	1		1	1						✓	✓	✓					1	
ATtiny87/167	20-32	8/16	0.512	10	11	✓			✓			1	2		1	1						✓	✓	✓		1 ⁽¹⁾			1	2
ATtiny261A/461A/861A	20-32	2-8	Up to 0.512	10	11	✓			✓						1	1	1					✓	✓	✓					1	1
ATtiny20x/40x/80x/160x	8-24	2-16	Up to 1	10	12	✓			✓			2			✓	1	1				✓	✓	✓	✓		1 ⁽¹⁾			1	1
ATtiny21x/41x/81x/161x/321x	8-24	2-32	Up to 2	10	12	✓		8	✓			2			✓	1	1				✓	✓	✓	✓		1 ⁽¹⁾			1	1
ATtiny441/841	14-20	4/8	Up to 0.512	10	12	✓		✓				1	2		1	2						✓	✓	✓		2			1	1
ATtiny1634	20	16	1	10	12	✓			✓			2	2		1	1	1					✓	✓	✓		2			1	1
ATtiny2313A	20	2	0.128	-	-	-			✓			2	2		1	1	1					✓	✓	✓		1			1	2
ATmega8A/16A/32A	28-44	8-32	1-2	10	8	✓			✓			2	1		✓	2	1					✓	✓	✓					1	1
ATmega8U2/16U2/32U2	32	8-32	0.5-1	-	-	-			✓			4	6		✓	2	3					✓	✓	✓		2			2	2
ATmega16U4/32U4	32	16/32	1/2	10	12	✓			✓			5			✓	1	1					✓	✓	✓		1	✓		1	1

Kuva 16. AVR-mikrokontrollereiden sisäiset toimintalohkot [7].

Analogisten signaaleiden muuntaminen digitaaliseen muotoon tehdään AD-muuntimella. Mikrokontrollerit sisältävät yhden AD-muuntimen, mutta ulkoisia tuloja on useita. Useista tuloista mitattava signaali ohjataan muuntimelle käyttäen sisäistä multiplekseriä.

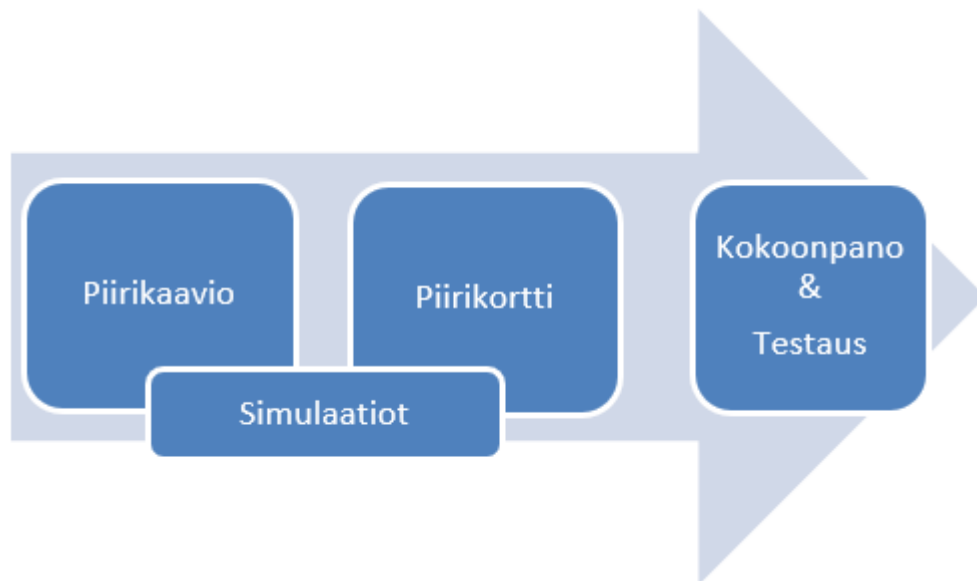
Ajastimet ja laskurit mahdollistavat aikariippuvien signaaleiden toteuttamisen. Näitä ovat esim. pulssinleveysmodulaatio (PWM), jota sovelletaan himmentimissä ja moottorien pyörimisnopeutta säädettyäessä.

Tiedonsiirtoon laitteen sisällä käytetään SPI-, TWI- tai 1-wire- väyläratkaisuja. Väylä mahdollistaa liittää samaan liityntään useita komponentteja. Väyläliityntäisiä ovat tyypillisesti digitaaliset anturit ja toimilaitteet laitekotelon sisällä. Ulkoinen tiedonsiirto toteutetaan sarjaporttia hyväksi käyttäen. Käytettäessä standardoitua RS232-tiedonsiirtoprotokollaa tarvitaan jännitetason sovitin TTL-logiikan ja RS232:n välille.

Muita mikrokontrollerin sisään rakennettuja toimintoja ovat sisäiset ja ulkoiset keskeytykset sekä virranhallintaominaisuudet (sleep mode).

4 Kehitystyökalut

Sulatettuun laitekehitykseen käytetään useita erilaisia suunnittelua helpottavia ohjelmistoja. Piirikaaviossa kuvataan komponentit ja niiden väliset kytkennät. Kytcentöjen toimivuus voidaan varmistaa simuloimalla. Suunnitteluohjelmistot toimivat yhteen, ja ne käyttävät yhteisiä symbolikirjastoja. Jos kytcentöjä muutetaan, niin muutos päivittyy koko suunnitteluketjuun (kuva 17).



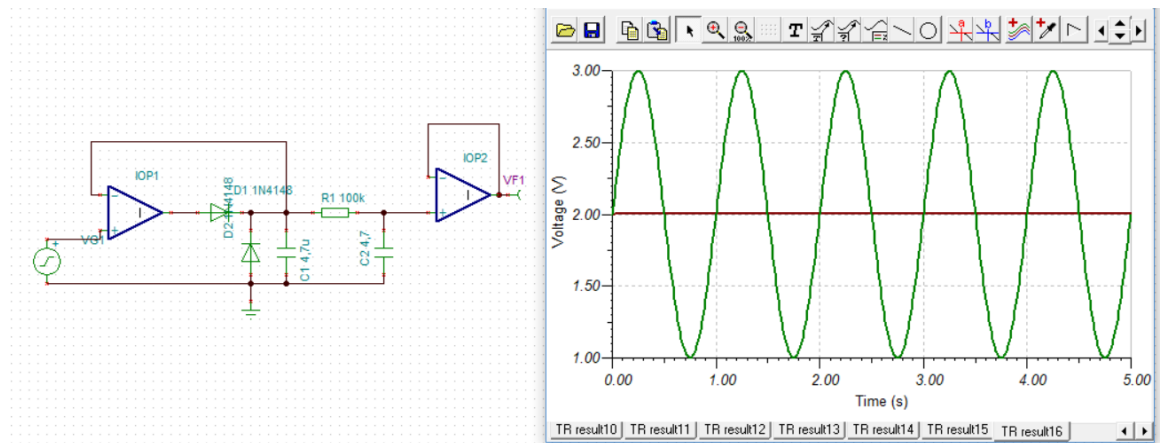
Kuva 17. Tyypillinen laitekehitysketju

4.1 Elektroniikkasuunnittelu

Elektroniikkasuunnittelu on tärkeä osa sulautettujen järjestelmien kokonaisuutta. Suunnittelussa korostuvat toimintavarmuus, luotettavuus ja kokemus. Laitesuunnittelun toimivuus varmistetaan testauksin ja erilaisten standardipohjaisten hyväksymismenettelyjen kautta [2]. Elektroniikkasuunnittelulla voidaan vaikuttaa laitteen luotettavuuteen ja kestävyys, ja myös komponenttivalinnoilla on merkitystä.

4.1.1 Simulaatiot

KytKentöjä ei aina kannata rakentaa koekytKentäalustalle, vaan ne voidaan simuloida ohjelmallisesti. Kokenut suunnittelija luottaa omaan ammattitaitoonsa ja noviisi voi kaivaa oppikirjat esille. Ohjelmallinen simulointi mahdollistaa nopean testaamisen erilaisilla komponenttikokoonpanoilla ja arvoilla. Useat komponenttivalmistajat tarjoavat vapaasti käytettäviä simulointiohjelmistoja (kuva 18). Simulaatio-ohjelmat mahdollistavat analogi- ja taikigitKentöjen simuloinnin.



Kuva 18. TINA Ti -ohjelmistolla simuloitu huippuarvon pitopiiri ja se analysoituna

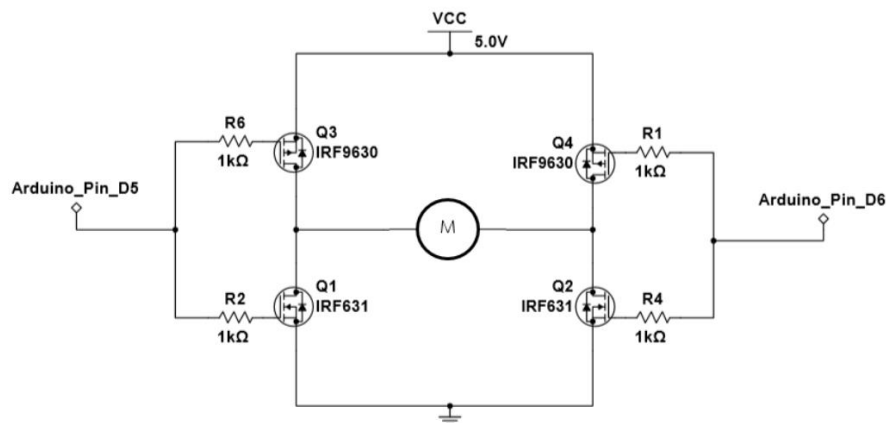
Usein käytettyjä simulaatio-ohjelmistoja:

- Texas Instruments, Tina-TI, analogisten kytKentöjen simulointi
- Texas Instruments, Webench, on suunnittelijan työkalu
- Mouser, MultiSIM, simulaatio, piirilevysuunnittelu ja komponenttihallinta
- Linear technology, LTspice

Käyttöönotto- ja käyttöohjeet löytyvät helposti esim. Youtube-videoina [8].

4.1.2 Piirikaavio

Elektroniikan osalta toteutus kuvataan piirikaaviossa. Piirikaaviossa kytkennät esitetään yksi toimintalohko omalla sivullaan (kuva 19). Sivujen väliset kytkennät esitetään käyttämällä signaaleissa nettinimiä, ohjelmisto osaa seurata sivujen välisiä.



Kuva 19. Piirikaavio H-silta

Komponenttien tekniset ominaisuudet ja esimerkkikytkennät löytyvät komponenttivalmistajan datalehdistä. Yleensä liityntä komponenttien ja kontrollerin välillä perustuu standardoituun rajapintaan, esim. SPI, i2c.

Analogiset tulot sovitetaan esiasteella mikrokontrollerin AD-muuntimelle optimaaliseksi. Esivahvistin vahvistaa signaalin ja optimoi mittausalueen 0 – 5 V:n välille.

Esim. huonelämpötilan mittausalue 10–35 °C

Lämpötila 10 °C vastaa jännitettä 0 V.

Lämpötila 35 °C vastaa jännitettä 5 V.

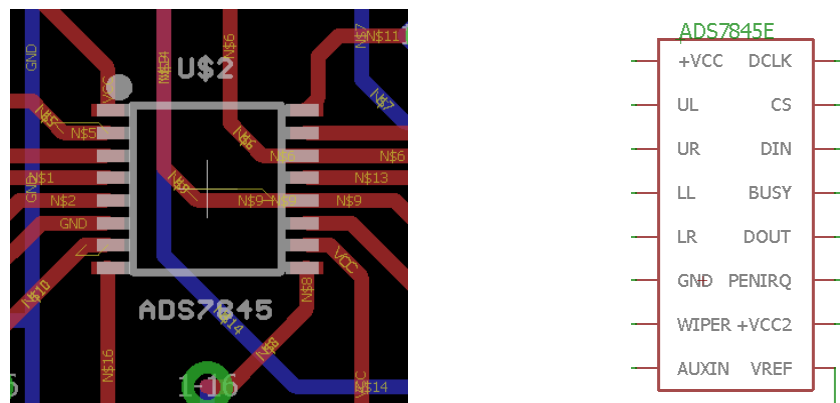
Näin saadaan paras mahdollinen mittaustarkkuus eli resoluutio.

Piirikaavion piirtoon on muutamia hyväksi havaittuja ohjelmistoja tai voidaan sanoa ohjelmistokokonaisuuksia. Ohjelmistot pitävät sisällään useita erillisiä ohjelmia, jotka käyttävät yhteisiä tie-

dostoja. Piirikaavio voidaan siirtää simulaattoriohjelmistoon (kuva 18), ja lopuksi piirikaavio linkitetään piirilevyn suunnitteluohjelmistoon. Ammattikäyttöön kehitetyt ohjelmistot ovat isoja tietokantapohjaisia kokonaisuuksia, esim. Menthor Graphisin PADS [8]. Oppilaitoskäytössä Autodeskin Eagle on saanut hyvää palautetta, ja opetuskäytössä ohjelmistosta on saatavissa ilmainen kolmen vuoden lisenssi. [10.]

4.1.3 Piirikortin suunnittelu

CAD-suunnitteluohjelmisto tukee piirikaavion (Schematic) piirtoa ja piirilevyn (Layout) suunnittelua. Ohjelmistot käyttävät yhteistä komponenttikirjastoa (kuva 20). Näin varmistutaan joustavan suunnittelun etenemisestä. Piirrosmerkkien symboliin voidaan sisällyttää tietoja esim. hankintapaikasta ja 3D-malli. Ohjelmistojen esittelyversiossa on usein rajoituksia kytkennän koon eli kytkentäpisteiden määrän tai piirilevyn koon suhteen.



Kuva 20. Komponentin fyysinen kuva ja piirikaaviosymboli

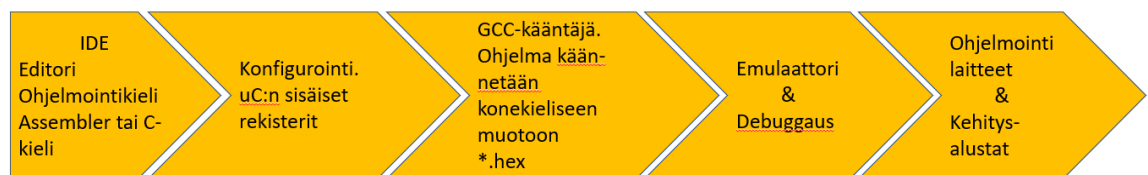
Suunnitteluprosessin alkuvaiheessa luodaan puuttuvat piirrosmerkit komponenteille. Piirrosmerkki tehdään piirikaavio- ja piirikortin suunnitteluun ja sen kytkentäpisteet tallennetaan.

Suunnittelun pohjaksi on yleensä tiedossa piirikortille varattu tila ja liityntäpisteiden paikat. Piirikortille sijoitetut komponentit voidaan johdottaa automaattisesti, mutta automaattijohdotus (autoroute) ei ole lopullinen, vaan suunnittelija joutuu viimeistelemään johdinreitityksiä ja johdinleveyksiä. Piirilevysuunnitteluohjelma tuottaa hankintadokumentation komponenttitoimittajalle

ja piirilevytehtaalle. Piirikorttien valmistusmäärästä riippuu kokoonpanon toteuttaminen. Pienet protosarjat voidaan juottaa perinteisin käsijuotosmenetelmin. Suuret sarjat ja paljon komponentteja sisältävät levyt kasataan automaattilinjastolla piirilevytehtaalla. Piirilevytehdas lisää kokoonpanoon suunnitteluohjelman tietoja ja esimerkiksi stensiilit juotospastan levitykseen.

4.2 Ohjelmistokehitys

Ohjelmistokehitystyökalut ovat mikrokontrollerikohtaiset. Piirivalmistaja tarjoaa työkalut ohjelmistokehitykseen (kuva 21). Markkinoilla on myös kolmannen osapuolen ohjelmistoja. Nämä työkalut ovat kehittyneempiä ja pitävät sisällään laajemman valikoiman toimintoja. Esimerkiksi IAR-systemsin workspens-ohjelmistossa on mikrokontrollerin alustukseen ns. velhotyökalu (wizard). Kolmannen osapuolen tarjoamat ohjelmointiympäristöt ovat hankintahinnaltaan useita tuhansia euroja, piirivalmistajan ohjelmistot ovat ilmaisia.



Kuva 21. Ohjelmistokehitystyökalut

4.2.1 Ohjelma-editori

Ohjelmakoodi kirjoitetaan tekstieditorissa, joka on läheistä sukua tekstinkäsittelylle. Kehittyneet editorit tarkistavat koodin syntaksin, kirjoitusasun sekä ennustavat komentoja. Projektin hallintatyökalu on osa koodieditoria, yhteys versionhallintaan on hyvä ominaisuus.

Editorin kautta hallitaan ja muokataan ohjelmistoprojektin ominaisuuksia ja toimintoja:

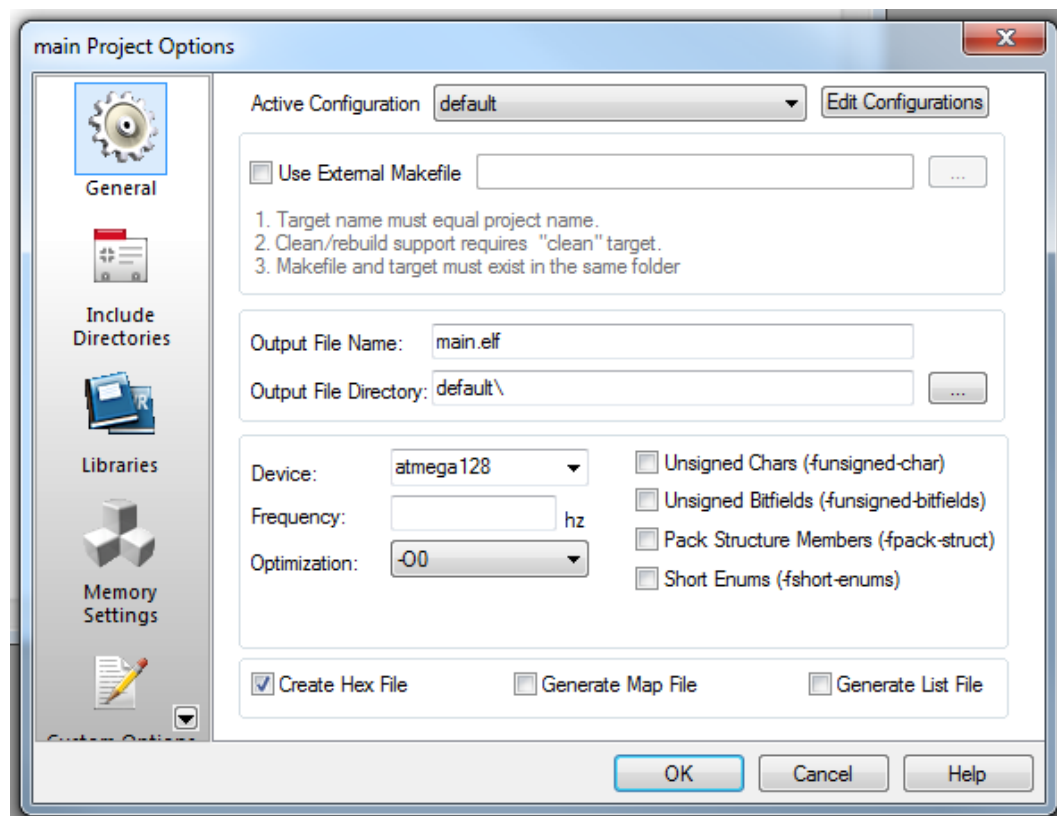
- Asetukset: Projekti ja mikrokontrolleri
- Kääntäjä
- Ohjelman lataus ja debuggaus

- Suorat linkit ohjekirjastoon

Projektiasetukset tallentuvat osaksi projektitiedostoa, kuvassa 22 on AVR-Studio asetusten muokkausikkuna. Valittu mikrokontrolleri sekä kääntäjän optimointi ovat tärkeimpiä valittavia kohteita. Mikrokontrollerin sisäiset asetukset voidaan muokata editorin kautta. Ohjelman testaaminen voidaan tehdä myös ilman kehitysalustaa suorittamalla ohjelma simulaattorissa.

4.2.2 Kääntäjä

Kääntäjä muuttaa C-koodin konekieliseen muotoon, konekielisen tiedoston pääte on HEX. Konekielinen ohjelma ladataan mikrokontrollerin FLASH-muistiin, josta se ladataan aina käynnistyksen yhteydessä. Konekielinen tiedosto on aina mikrokontrollerikohtainen, ja sitä ei voi käyttää muiden mikrokontrollereiden kanssa.



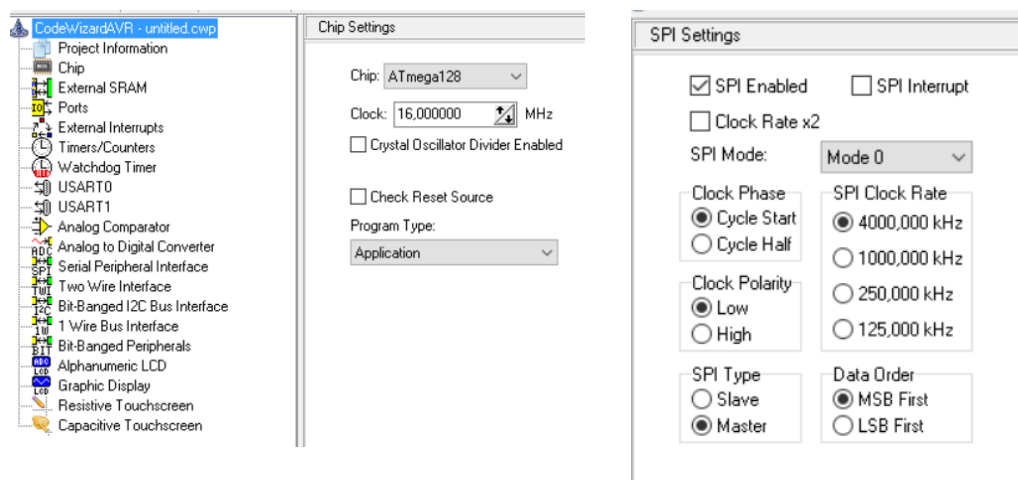
Kuva 22. Kääntäjän asetukset AVR-studiossa

Käännöksen parametrit tallentuvat projektitiedostoon. Kääntäjän optimointiin liittyviä parametrejä ovat esimerkiksi:

- Debug- / release käännös
- Kääntäjän lupa optimoida koodia.

Ohjelmoijan kannattaa tarkkailla muistin käyttöä aina käännöksen yhteydessä, koska mikrokontrollerin ohjelmamuisti on rajallinen.

Eräät ohjelmakehitysympäristöt pitävät sisällään koodigeneraattorin, joka luo mikrokontrollerin koodin rungon ja alustuksen haluttujen parametrien mukaan. Suunnittelija valitsee käytettävän mikrokontrollerin ja sen kellotaajuuden. Seuraavaksi valitaan sisäiset toimintalohkot ja valitaan tarkentavat määrittelyt (kuva 23).



Kuva 23. AVR-kontrollerin alustus Code Vision -ohjelmistossa.

Lopuksi generoidaan asetusten mukainen ohjelmakoodi. Tuotettu ohjelmakoodi perustuu mikrokontrollerin sisäisiin rekistereihin ja on siirrettävissä toiseen ohjelmointiympäristöön.

4.2.3 Ohjelman lataus ja virheenkorjaustyökalut

Tuotantoprosessin aikana mikrokontrollerit ohjelmoidaan valmiiksi ennen kokoonpanoa, ja tällöin käytetään erillistä ohjelmointilaitetta. Ohjelmointilaitte tallentaa ohjelman mikrokontrollerin Flash-muistiin ja muokkaa piirin sisäiset asetukset halutuiksi.

Kehitysalusta on ohjelmiston kehitystyökalu, jonka avulla koodia testataan todellisessa mikrokontrolleri ympäristössä. Oikeat liitynnät on rakennettu kehitysalustan ympärille käyttäen koe-kytkentäalustaa ja muita väliaikaisia kytkentöjä. Kehitysalustassa on kiinteä ohjelmointiliityntä, vaikkakin se varaa mikrokontrollerista muutamia liityntöjä. Esimerkiksi käytettäessä ATmega128-mikrokontrolleria Jtag-liityntä tarvitsee 4 liityntää, jotka ovat AD-muuntimen tulot 4–7. Ohjelmointiliityntää käytetään ohjelmoinnin lisäksi ohjelmakoodin testauksessa.

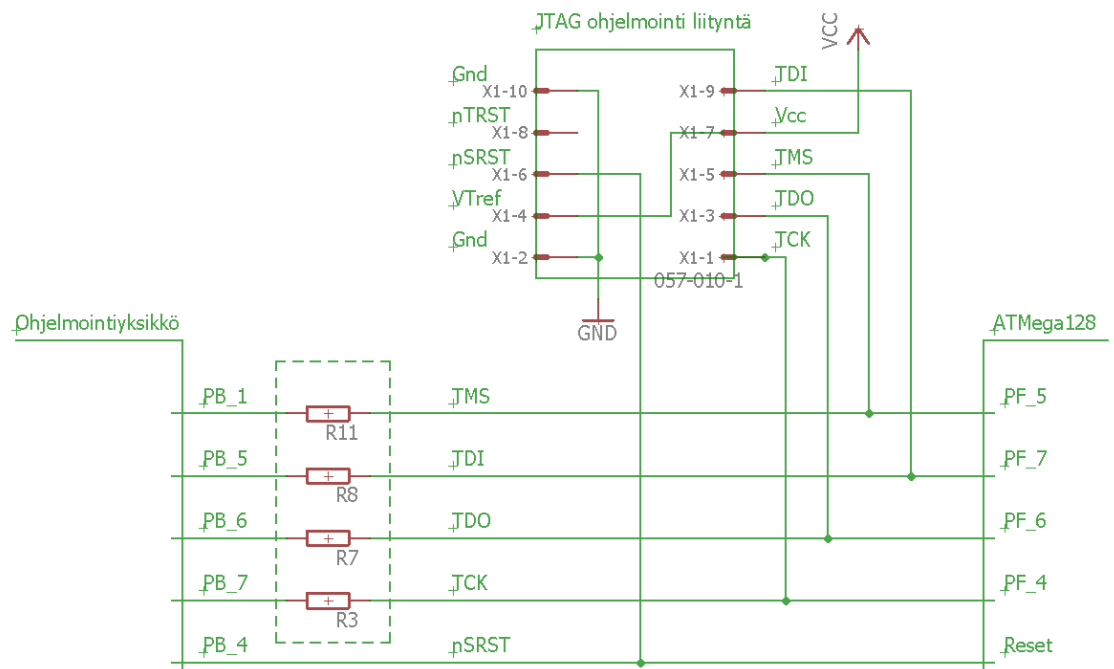
Taulukko 3. Microchip-kontrollerien ohjelmointiliityntä

	UPDI	TPI	SPI	debug- WIRE	JTAG	PDI	aWIRE	SWD
AVR Tiny	Uudet	Osa	Osa	Osa				
AVR Mega				Osa	Osa			
AVR X Mega					Osa	Kaikki		
AVR UC					Kaikki		Osa	

SAM	Osa	Kaikki
-----	-----	--------

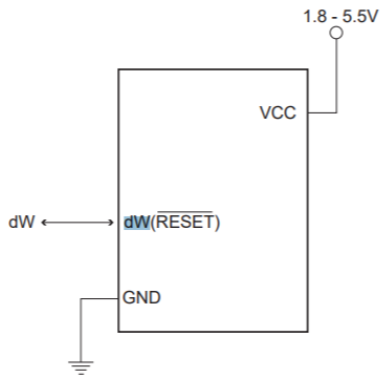
Käytettävä ohjelmointiliityntä riippuu valitusta mikrokontrollerista. Taulukossa 3 on ohjelmointiliityntöjä ja niitä tukevia Microchip-mikrokontrollereita. Ohjelmointiympäristöissä ohjelmalataus- ja testaustoiminnoista käytetään nimitystä debuggaus. Debuggaustilassa kehitettävää ohjelmaa voidaan ajaa käsky kerrallaan ja muuttujien arvoja ja niiden muutoksia voidaan seurata. Editorissa ohjelmaan voidaan lisätä pysäytyspisteitä (Breakpoint), joissa ohjelman suoritus pysähtyy.

Jtag-liityntä on standardisoitu IEEE1149.1:ssä [11], joka määrittelee Jtag-liitynnän ja kytkennän mikrokontrollerin ja ohjelmointilaitteen välillä (kuva 24). Jtag-liityntä mahdollistaa ohjelmiston lataamisen mikrokontrollerin FLASH-muistiin ja testaamisen kehitysvaiheessa. Mikrokontrollerin muistiin ladattua ohjelmaa voidaan ajaa yksi käsky kerrallaan ja muuttujien arvoja voidaan seurata koodieditorissa. Jtag-liityntä mahdollistaa koodin laitteiston testaamisen tuotanto- ja loppu-testausvaiheessa.



Kuva 24. Jtag-liityntä ja liitin

DebugWIRE on käytössä mikrokontrolleriperheen pienimmissä piireissä (kuva 25). DebugWIRE soveltaa Jtag-liityntää, useat ohjelmointilaitteet toteuttavat liitynnän erillisellä adapterilla (taulukko 4).

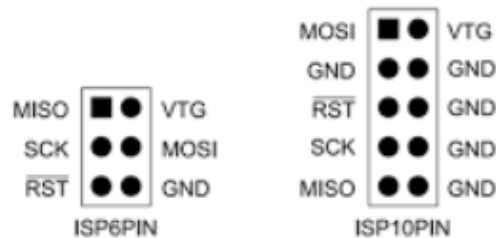


Kuva 25. ATTiny13 dWIRE-liityntä

Taulukko 4. Jtag vs. aWire liitinjärjestys

Jtag liitinjärjestys	Signaali	aWIRE liitinjärjestys
1: TCK		
2: Gnd	Gnd	6
3: TDO	Data	1
4: VTG	VTG	2
5: TMS		
6: nTRST		
7: vapaa		
8: nTRST		
9: TDI		
10: Gnd		

In-System Programming (ISP) on sarjamuotoinen ohjelmointiliityntä, liityntää tukevat ARM Prosessorit, AVR PIC ja PIC mikrokontrollerit. ISP on ohjelmointiliityntä, ja se ei mahdollista ajonai-
kaista debuggausta. Liitynnöissä ovat 10- ja 6-pinniset versiot (kuva 26).



Kuva 26. ISP-ohjelmointiliitynnät.

Tänä päivänä ovat yleistyneet nettipohjaiset ohjelmointiympäristöt; selaimen kautta käännetty ohjelma kopioidaan kehitysalustan muistiin. Kehitysalustan ohjelmamuisti näkyy käyttäjälle ulkoisena levyasemana, minne ohjelman uusi käännösversion kopioidaan. Kehitysalusta käynnistyy uudelleen ja ottaa viimeisen version käyttöönsä.

Sulautetun ohjelman kehityksessä tärkeää roolia näyttelee ohjelmiston debuggausominaisuus. Debuggaus mahdollistaa muuttujien arvojen seuraamisen ja muuttamisen ohjelman suorituksen aikana. Debuggaus helpottaa ohjelmiston toiminnan ymmärtämistä, sillä ohjelman eteneminen voidaan pysäyttää epäselvään kohtaan ohjelmakoodia. Virhetilasta toipumista voidaan testata pysäyttämällä ohjelma ja "sotkemalla" muuttujien arvot ja katsomalla, kuinka tämä vaikuttaa lopputulokseen.

Liityntää käytetään tuotannossa laitteiden automaattiseen lopputestaukseen. Ulkoinen laite ohjaa Jtag-liitynnän kautta liityntöjä eri tiloihin, joiden vaikutus tarkistetaan ulkoisista liitynnöistä.

Jtag-liitynnän käyttö vaatii ulkoisen ohjelmointilaitteen käyttöä. Yleisimpien kehitysympäristöjen tukemat laitteiden hinnat ovat noin sata euroa (Atmel ICE, ICE II, Dragon) (kuva 27). Kolmannen osapuolen tarjoamat laitteet ovat edullisempia: Olimex AVR-MT-128 maksaa noin 20 euroa.



Kuva 27. Atmel ICE ja ICE II debuggauslaite

4.3 Testaus

Luotettavan järjestelmän toteuttaminen edellyttää huolellista laitteiston ja ohjelmiston testausta. Kehitettävälle järjestelmälle luodaan testaus suunnitelma jo suunnitteluvaiheessa. Testaus suunnitelma ottaa kantaa, kun laaja testaus- ja hyväksymisprosessi on tarpeen toteuttaa. Laajuuteen vaikuttavat laitteiston käyttökohde ja käyttökohteessa sovellettavat testaus standardit. Sovellettavien standardien lisäksi laitteiston tilaaja voi antaa omat testauksen standardeja tiukemmat raja-arvot, ja tämä on yleistä ainakin ajoneuvovalmistajien kanssa toimittaessa. Testauksesta tehdään aina kirjallinen dokumentaatio. Versionhallintatyökalut ovat suuri helpotus tässäkin suhteessa.

Testaus standardit eri toimialoilla [2]:

- Kotitalous: EN 61000-6-1/3
- Teollisuus: EN 61000-6-2/3
- Sairaala: EN 61000 -6-1/2
- Ajoneuvo: ISO 16750 -2

Standardit määrittelevät testaukset toimialan mukaan. Laitteiden toiminta ja luotettavuus varmistetaan virallisessa laboratoriossa ennen myynnin aloittamista. Laboratorio antaa testauksesta

kirjallisen dokumentaation. Testauksen yhteydessä havaitut ongelmat saatetaan tuotekehityksen kehityksen tietoon, jossa ongelmat ratkaistaan ja testaamista jatketaan.

4.3.1 Ohjelmistotestaus

Ohjelmistotestaus kattaa koko tuotekehitysprosessin. Testauksesta tehtävät dokumentit sekä niiden määrä ja laajuus riippuu projektista. Minimissään ohjelmakoodin testaus dokumentoidaan koodiin, ja kattavana kommentointi helpottaa koodin lukemista ja ymmärtämistä.

Ohjelmistoa voidaan testata kehitysalustalla ennen lopullisen elektroniikkaosuuden valmistamista. Ohjelmakoodia ajetaan ns. Debug-tilassa, missä on mahdollisuus simuloida erilaisia syötteitä ja luoda vikatilanteita, joista laitteiston tulee toipua. Jtag-liityntä on erinomainen apu, ja se mahdollistaa jopa automaattisen testausjärjestelmän käytön. Kaikki ohjelman hierarkiatasot testataan ja dokumentoidaan

Ohjelmistotestauksella pyritään varmistamaan:

- Toimintalogiikka, koodin toiminta.
- Mikrokontrolleri (rekisteritasolla)
 - liityntäpiirien rajapinnan toiminta, suuntarekisterit
 - ohjelmamuisti ja sen riittävyys.
- Tiedonsiirto
 - nopeus, kättelyt, muuttujatyypit.
- Oheislaitteet
 - UI, näyttö ja näppäimistö
 - muisti SD-kortti.

4.3.2 Elektroniikkatestaus

Visuaalinen testaus tehdään elektroniikkakomponenttien kokoonpanon jälkeen ennen sähköistä testausta. Visualisessa tarkastuksessa varmistutaan esim. komponenttien asennus; komponenttien tulee olla tarkasti oikein sijoittuneita ja tyyppin täsmälleen oikein. Visuaalinen testaus on mahdollista automatisoida AOI-laitteilla. AOI perustuu kamera- ja tekoälypohjaiseen järjestelmään, sen etu on nopeus.

Sähköisellä testauksella varmistutaan kytkentöjen oikeellisuus ja toimivuus. Ensimmäisessä vaiheessa varmistetaan syöttöjännitteen raja-arvot ja virrankulutus. Usein laitteeseen ladataan erillinen testausohjelmisto tai testaus voi olla osa järjestelmää. Testaus aktivoidaan, ja laitteisto suorittaa sisäisen testisekvenssin ja antaa lopuksi testituloksen.

Tuotantovaiheessa vialliset laitteet ja komponentit merkitään ja poistetaan tuotannosta.

4.3.3 Hyväksymistestaus

Hyväksymistestaukset tehdään akkreditoidussa laboratoriossa. Näin varmistutaan, että mittaus- tulokset ja menetelmät ovat vertailukelpoisia kansallisesti ja kansainvälisesti. Testausprosessin hyväksytysti läpäissyt laite saa todistuksen ja luvan käyttää CE-merkintää (kuva 28).



Kuva 28. CE-merkintä

Laitevalmistaja ilmoittaa laitteen tyyppikilvessä CE-merkinnällä, että laite täyttää sille asetetut vaatimukset [12]. Kuvassa 29 on terveydenhoitokäyttöön hyväksytty laite. Sen tunnistaa CE-merkinnän yhteydessä olevasta numerokoodista 0459.



Kuva 29. Terveystenhoitoon hyväksytty laite

Laitetestaus varmistaa sähkömagneettisen yhteensopivuuden muiden laitteiden kanssa. Tämä tarkoittaa, että laitteisto ei lähetä häiriöitä ulospäin eikä häiriinny muiden laitteiden lähettämästä häiriöstä. Laitteisto ei saa myöskään häiriintyä sisäisistä häiriöistä. Tällaisia voisivat olla esimerkiksi maadoitusongelmat [12].

4.3.4 Tuotantotestaus

Kokoonpanon yhteydessä laitteisto testataan ennen varsinaista luovutusta asiakkaalle. Toiminta varmistetaan suunnitellulla proseduurilla ja erillisellä testerillä. Toimintojen testauksen lisäksi tarkistetaan sisäinen ohjelmaversio ja mahdollisesti lukitaan mikrokontrolleri uudelleen kirjoitussuojabiteillä.

Vaativiin olosuhteisiin suunniteltu laitteisto käy läpi vanhennusprosessin ennen lopputestausta. Vanhennuksessa laitteistoa rasitetaan eri lämpötiloissa useita vuorokausia. Lopuksi laitteisto viritetään ja varmistetaan kokonaisen laitteiston luotettava toiminta koko toiminta-alueella. Lopputestauksen yhteydessä laite merkataan ja testauspöytäkirja arkistoidaan.

5 Kehitysalustan toteuttaminen

Kehitysalustan määrittely tehtiin yhdessä Kajaanin ammattikorkeakoulun opettajien kanssa, jotka olivat käyttäneet aikaisempaa 8051-kehitysympäristöä. 8051-kehitysympäristö oli tullut tiensä päähän, ohjelmistotuki oli päättynyt eikä se toiminut uusissa Windows-työasemissa. Uusissa työasemissa ei ollut sopivaa rinnakkaisliityntää ilman erillistä laitehankintaa, ja sopivan laitekoko-panon löytäminen oli vaikeaa.

Suunnitteluprosessin alussa tehtiin katselmointi muiden oppilaitosten käyttämiin ja kaupallisiin opetusympäristöihin. Yleensä oppilaitokset käyttivät sulautettujen järjestelmien opetuksessa kaupallisia kehitysalustoja, mutta löysimme myös muutamia omia toteutuksia.

HS-Heilbronn UAS [13] oli samantyyppinen sulautettujen järjestelmien opetukseen käytettävä kehitysalustan suunnitteluprojekti nimeltään uECU (kuva 30). Heidän kanssaan vaihdettiin mielipiteitä ominaisuuksista, joita kannattaa ottaa huomioon suunnittelussa. Keskusteluissa totesimme yhteisesti seuraavaa:

- Edullinen, voidaan antaa oppilaiden käyttöön.
- Yleisesti käytetty mikrokontrolleri
- Näyttö ja näppäimet kortilla
- Vapaasti käytettävät ohjelmistotyökalut.



Kuva 30. uECU-kehitysalusta

HS-Heilbronn UAS:n (Saksa) uECU- kehitysalustan (kuva 30) tekniset tiedot:

- Mikrokontrolleri: AtMega88-kannalla
- 2-rivinen Lcd-näyttö
- NTC-lämpötila-anturi
- LDR-valovastus
- 2 kpl ledejä ja 4 kpl kytkimiä.

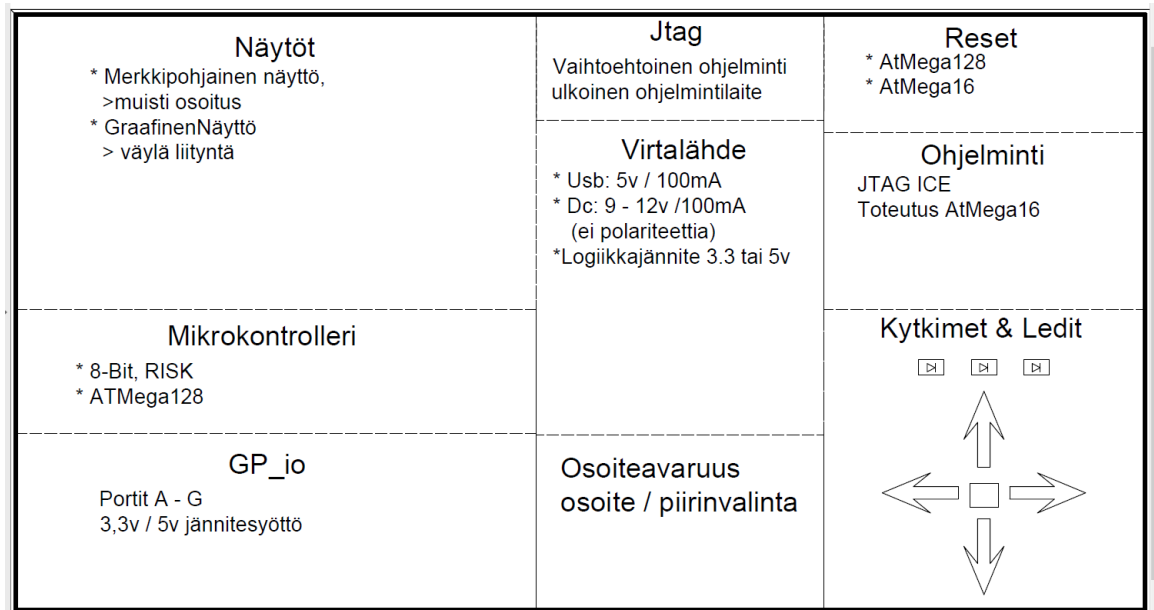
Seinäjoen AMK:ssa on kehitetty kehitysalusta Surfnet [14]. Kehitysympäristö on suunniteltu lähinnä langattomien mittaussovellusten alustaksi. Surfnet on kehitetty kansallisella projektirahoituksella, ja suunnittelussa on ollut mukana useita alan toimijoita, mikä on tehnyt järjestelmästä laajan. Surfnet-ympäristössä on kiinnitetty erityistä huomiota dokumentoinnin tasoon ja laatuun.

Opetuskäytössä käytettävän laitteen tulee olla helppokäyttöinen ja kestävä. Elektroniikkaa käsitellään usein ilman ESD-suojauksia, ja mittalaitteiden mittapäät aiheuttavat oikosulkuja kytkennöissä. Luotettavuuteen pitää paneutua jo suunnitteluvaiheessa ja maadoitukset ovat tärkeässä roolissa.

Ulkoasun tulee olla selkeä ja johdonmukainen. Elektroniikkakortti on mitoitettava EURO-standardin mukaan: ulkomitat 100 x 160 mm (EURO 3U). Liittimet ryhmitellään samalla kaavalla, pinni 1 vasemmalla alhaalla. Kaikki liittimet merkitään selkeästi:

- Nimi
- Pinni 1

Kuvassa 31 on esitetty kortin lohkokaavio, johon hahmoteltiin toimintalohkoja ja niiden sijoittelua.



Kuva 31. Kamk-opetuskortin lohkokaavio

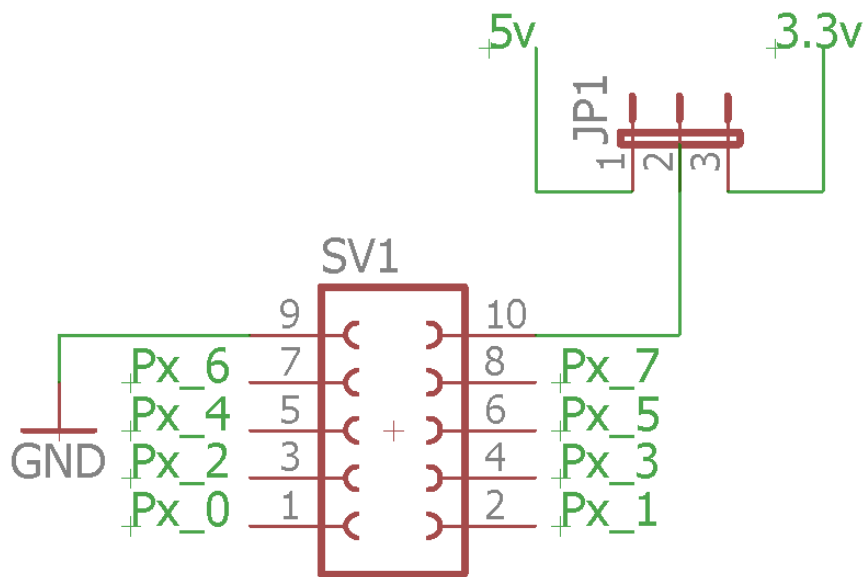
Kajaanin ammattikorkeakoulussa päädyttiin seuraavaan kokonaisuuteen.

Mikrokontrolleri:

Kehitysalustaan valittiin yleiskäyttöinen Microchipin valmistama ATMega128L. ATMega 128 perustuu samaan kahdeksanbittiseen RISC-arkkitehtuuriin kuin aikaisemmin opetuskäytössä ollut 8050. Kontrollerissa on kattava valikoima sisään rakennettuja toimintalohkoja. Samasta piiristä on saatavilla oma versio ajoneuvo- ja USB-sovelluksiin. Ohjelmamuistia on 128 kB, joka riittää hyvin opetuskäytössä. Kontrolleri toimii laajalla käyttöjännitealueella (2,7–5,5 V), ja logiikkajännitetasot ovat valittavissa 3,3 V tai 5 V (TTL-taso). Valmistajan dokumentaatio on erittäin kattavaa ja käyttäjäryhmiä on useita. Dokumentaatiosta löytyy eritasoisia ohjeita ongelmatilanteisiin.

GPIO (General Purpose I O):

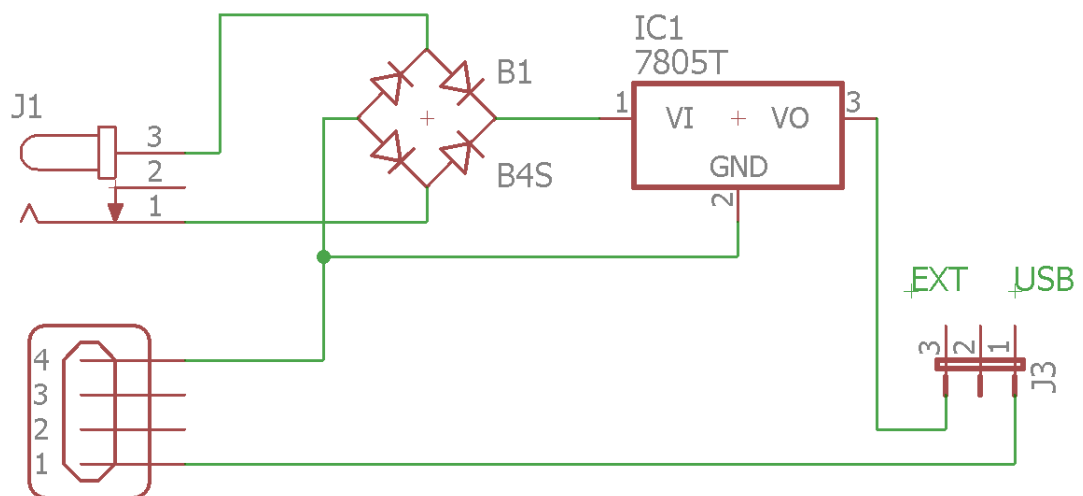
Kontrollerin kaikki erikseen ohjelmoitavat IO-pinnit ovat liittimillä ja niitä on 5 kpl, liityntöjen ryhmittely 8 bitin kokonaisuuksiin, portit A–G. Kuvassa 32 porttiliityntäkytkentä ja käyttöjännitetasot omia kytkentöjä varten, käyttöjännite on valittavissa jumpperilla 5 V tai 3,3 V.



Kuva 32. GPIO-porttikiytentä

Virtalähde

Kehitysalusta saa tarvittavan käyttöjännitteen ulkoisesta virtälähteestä DC-pistokkeen kautta tai USB-ohjelmointiliittimestä (kuva 33). Ohjelmankehitysvaiheessa opetuskortti saa käyttöjännitteen USB-B-ohjelmointiliittimen kautta. Opetuskortin lisäksi erillistä virtalähdettä ei siis tarvita, vaan kortti saa sähkönsä työaseman USB-portista.

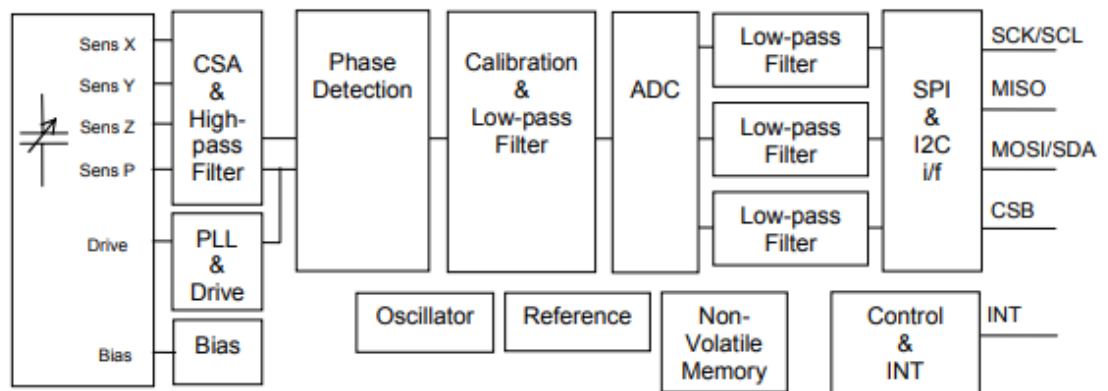


Kuva 33. Virtalähdevalinta ja tasasuuntaus

Opetuskortin irrottaminen tietokoneesta esim. testausta varten onnistuu kytkemällä käyttöjännitteet erillisestä DC-pistokkeesta 8–12 V. Tasajännitesyöttö on suojattu väärin kytkennältä, suojaus on toteutettu tasasuuntaussiltaa hyödyntäen. DC-liityntään voidaan liittää akku tai ulkoinen tasajännitelähde.

Kiinteät anturit

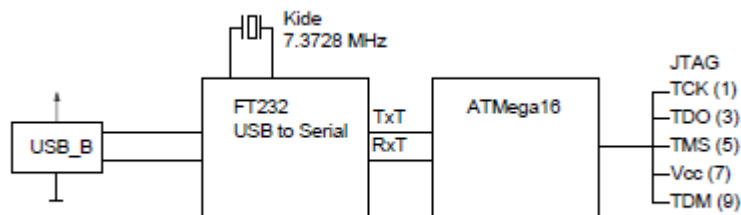
Kehitysalustalle suunniteltiin kiinteiden kytkimien ja ledien lisäksi kiihtyvyyssanturi. Kiihtyvyyssanturiksi valittiin 3-akselinen VTI Technologiesin CMR3000. Se on edullinen, ja sitä käytetään useissa pelikonsoleissa. Kiihtyvyyssanturia hyödyntämällä on mahdollista tutustua SPI-väylän toimintaan laboraatioharjoitusten avulla. Lisäksi anturi mahdollistaa pienten pelisovellusten kehittämisen. Anturissa on vaihtoehtoinen IIC- tai SPI- liityntä (kuva 34).



Kuva 34. CMR3000 3-akselinen kiihtyvyyssanturi [15]

Ohjelmointi, Jtag-liityntä

Opetuskortille päätettiin erillinen toteuttaa ohjelmointiyksikkö, joka pohjautuu ATmega128:n Jtag-liityntään. AVR-käyttäjäryhmässä on julkaistu kytkentä ja ohjelma ohjelmointilaitteesta, joka on yhteen sopiva kaupallisen Atmel ICE -laitteen kanssa. Kuvassa 35 on ohjelmointilohkon lohko-kaavio.



Kuva 35. Ohjelmointiyksikkö – lohkokaavio

Ohjelmointiyksikkö on toteutettu ATmega 16 -mikrokontrollerilla ja USB-sarjaliikennemuuntimen avulla. Mikrokontrolleri ja sarjaliikennemuunnin käyttävät ulkoista kidettä. Tästä syystä piiri täytyy ohjelmoida ennen piirikortille juottamista.

Reset

Opetuskortilla on kaksi erillistä Reset-linjaa, ohjelmointi ja varsinaisen mikrokontrollerin käyttämä. Molemmilla on pienoiskytkimet resetointia varten.

Väylä- ja osoiteavaruus

Opetuskortilla on osoitedekoodaukseen erilliset liittynät Adress A0–A7 ja piirinvalinta /CS0 ... /CS7. Esimerkin omaisesti, merkkipohjainen näyttö on kytketty osoiteavaruuteen.

Näytöt

Kortille haluttiin perinteinen merkkipohjainen ja graafinen näyttö. Molemmille näytöille on valmiit liittimet. Merkkipohjainen näyttö vaatii ulkoisen kontrastinsäätökytkennän, joka toteutettiin kortin laitaan omalla liittimellä. Merkkipohjainen näyttö on kaksirivinen, ja rivillä on 16 merkkiä. Näytöt perustuvat HD44780-näytönohjainpiiriin (kuva 36).



Kuva 36. Merkkipohjainen näyttö

Graafinen näyttö mahdollistaa monipuoliset käyttöliittymä- ja pienpelisovellukset. Näytöksi valittiin Batron BTHQ128064AVC1 (Kuva 37), näytön resoluutio on 128 x 64 pikseliä.



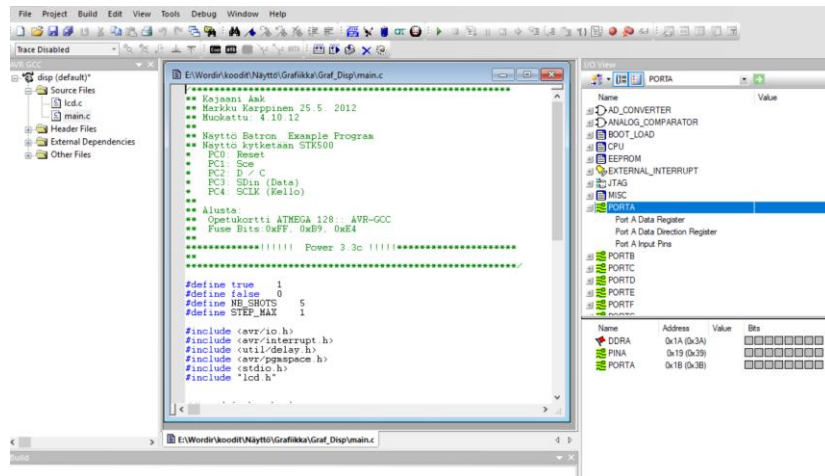
Kuva 37. Batron 128 x 64 graafinen näyttö

Näyttö perustuu UltraChip UC1606 -näytönohjaimeen, joka on integroitu näyttöön. Näytössä on myös LED-taustavalo. Näytön liityntä on kytketty mikrokontrollerin TWI-väylään.

Ohjelmointiympäristö AVR Studio 4.18

ATmega128-mikrokontrollerille oli saatavana useita ohjelmointiympäristöjä. Opetuskäyttöön valittiin valmistajan sivuilta vapaasti ladattavissa oleva versio, jossa on c-kielinen kääntäjä. Kääntäjä on WINAVR, joka on GNU-pohjainen, ja se integroituu AVR-ympäristöön asennusvaiheessa automaattisesti. Opetuksessa käytetään AVR-Studio version 4.18 (kuva 38), koska se tukee toteutettua Jtag-liityntää. Peruseditori ei sisällä nykyaikaisia ominaisuuksia, ja sen viimeisin versio on julkaistu 2009. Ohjelmistosta löytyvät mm. seuraavat ominaisuudet:

- projektin hallinta
- ohjelmaeditori ja ohjekirjasto (Help)
- debugaus
- mikrokontrollerin rekisterihallinta
- simulaattori
- kontrollerin sisäisten (FUSES) asetusten käsittelytyökalut



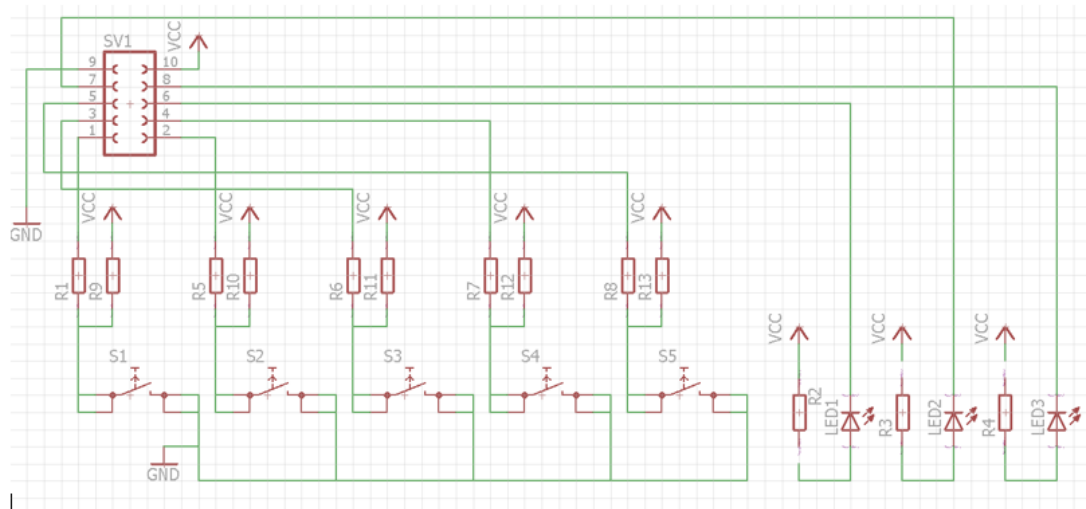
Kuva 38. AVRStudio v4.18 käyttöliittymä.

Kytkimet & ledit

Kortilla on LED:t 4 kpl ja kytkimet 5 kpl, ja niitä ei ole kytketty kiinteästi mikrokontrolleriin, vaan niillä on oma liitin (SW&LEDS). Ledit ja kytkimet otetaan käyttöön kytkemällä lattakaapelilla haluttuun porttiin (kuva 39).

Ledit ja kytkimet ovat nolla-aktiivisia:

- Led palaa, kun lähtö on tilassa "0".
- Kytkimet toimivat vapaana, tulo on "1" ja painettuna "0".



Kuva 39. Kytkimet & ledit

5.1 ATMega128

Mikrokontrolleriksi valittiin yleiskäyttöinen Atmelin suunnittelema ja Microchipin valmistama AT-Mega 128. ATMega128 on 8-bittinen RISC-pohjainen mikrokontrolleri, jolla on pientä historiallista yhteyttä aikaisempiin 8051-kehitysympäristöihin. Kontrollerista on saatavilla rinnakkaisversioita, joiden liityntärajapintaa on laajennettu:

- AT90CAN128 Can-sovellukset
- AT90USB128 USB-sovellukset.

Kontrollerissa on kattava kokoelma sisäisiä toimintalohkoja (Peripheral Features):

Ohjelmamuisti (Flash): 128 kB

Datamuisti (RAM): 4096 bittiä

Tiedonsiirto: 2-sarjaporttia, SPI, I2C

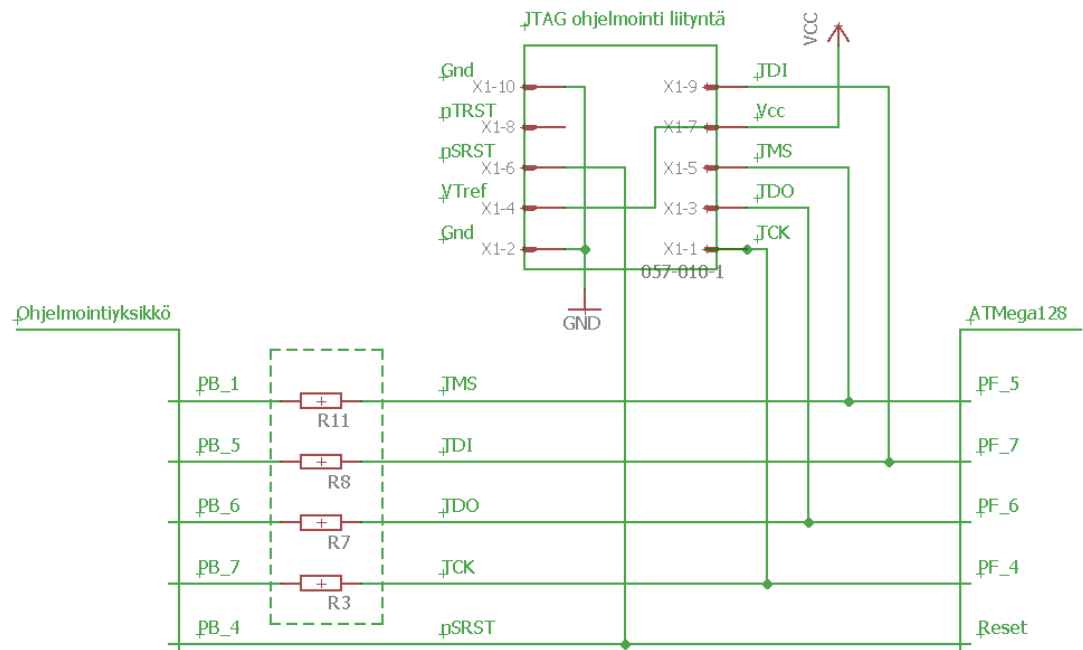
Ajastin (Timer): 2x8-bit, 2x16-bit

AD-muunnin: 8 kpl 10 bit tuloa

GPIO: 53 kpl (6 kpl 8 bit)

Kontrollerin ohjelmointi ja ohjelmistotestaus (debuggaus) toteutettiin Jtag-rajapintainen ohjelmointiyksikkö. Ohjelmointiyksikössä on oma mikrokontrolleri (ATMega16) ja ohjelmisto, Jtag-yksikköä ei suunniteltu itse, vaan se kopioitiin käyttäjien keskustelufoorumilta: Aquaticus Jtag mk1.

Jtag-osio on kiinteästi kytketty mikrokontrollerin Jtag-liityntään. Kytkeä poistaa käytöstä AD-muuntimen neljä ylintä tulokanavaa: ADC5 ... ADC8. Ohjelmointiyksikön ja kontrollerin välille on lisätty standardin mukainen Jtag ulkoista ohjelmointilaitetta varten. Ulkoisen ohjelmointilaitteen käyttö mahdollistaa AVR Studion uusien versioiden käytön. Uudet versiot eivät tue kortilla olevaa Jtag-ohjelmointiyksikköä (Jtag_ICE).



Kuva 40. Jtag-liityntä & ohjelmointiyksikkö

Käytettäessä ulkoista Jtag-liityntää (kuva 40) poistetaan vastukset R3, R7, R8 ja R11.

Kontrollerin kellotaajuus saadaan ulkoiselta kiteeltä (liittimellä) tai kontrollerin sisäisestä oskillaattorista, jonka kellotaajuutta voidaan asetuksilla muuttaa välillä 1–8 Mhz.

Microchip tarjoaa selkeät manuaalit (datakirjat), jotka sisältävät kattavan dokumentaation sisältäen esimerkkejä Assembler- ja C-kielellä.

5.2 Suunnitteluprosessi

Oppilaitoksessa on käytössä PADS-suunnitteluohjelmisto ver 9.3.0 [9], ja varsinainen suunnittelu-prosessi vietiin läpi sen avulla.

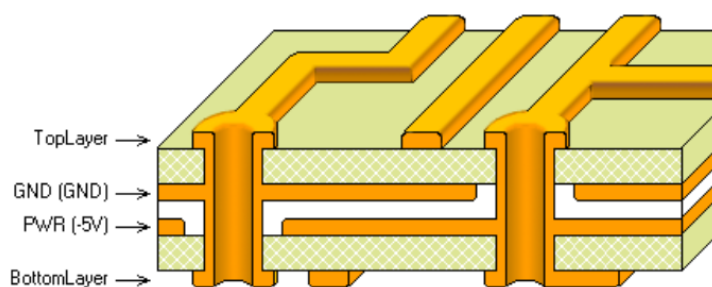
- piirikaaviosuunnittelu (Schematic)
- piirilevysuunnittelu (Layout)

Piirikaaviosuunnittelu tehtiin PADC Schematic -ohjelmistolla. Piirikaaviossa jokainen suunnitelman osa dokumentoidaan omalla sivulleen, esim. virtalähde, mikrokontrolleri, liittynät. Kytkenän osa liitetään yhteen yhteisillä nettinimillä, esim. GND, VCC_5, VCC_3, Jtag_TMS... Ohjelmisto kykenee seuraamaan kytkentää nettinimien avulla, ja aktivoimalla nettiohjelma näyttää sivut, missä kyseinen netti esiintyy.

Piirikortinsuunnittelu tehtiin Layout-ohjelmistolla, joka hyödyntää piirikaavion kytkentätietoja. Mahdolliset muutokset päivittyvät automaattisesti molempiin dokumentteihin. Piirikortin suunnitteluun vaikuttavat useat tekijät:

- Mekaaniset mitat ja muodot
- Komponenttien ja liittimien sijoittelu

Kortin luotettavuuteen vaikuttaa oleellisesti kortin maadoituksen suunnittelu ja toteutus. Komponenttien sijoittelun edetessä alkoi näyttää selvältä, että kortille tarvitaan useampia johdinkerroksia kuin kaksi. Korttiin suunniteltiin neljä johdinkerrosta (kuva 41).



Kuva 41. 4-kerrospiirilevy. [9]

- 1- Päällimmäinen taso, lähinnä mikrokontrolleriympäristö
- 2- Käyttöjännitteet
- 3- Maataso
- 4- Pohjataso, ohjelmointiyksikkö

Yhtenäinen maataso on luotettavuuden kannalta hyvä ratkaisu, koska suunnittelussa ei tarvitse huolehtia maadoituslenkeistä.

5.2.1 Piirikortin toteutus

Piirikortin mitoitukselle ei määritelty tarkkoja raja-arvoja. Käytännössä kaikkea ei ole tarkoituksenmukaista ahtaa mahdollisimman pieneen tilaan. Käytettävyyden kannalta päätettiin tehdä standardin mukainen EURO 3U (100 x 160 mm) piirilevy. Piirikortin yläpuolelle sijoitettiin käytettävyyden kannalta tärkeät komponentit. Alapuolella on JTG-ohjelmointiyksikkö ja sen tarvitsema USB-sarjaliikennemuunnin kytkentöineen.

Komponentit sijoitettiin lohkoakaavion (kuva 31) mukaisesti. Tämän jälkeen käytettiin ohjelman Autoroute-toimintoa. Automaattinen johdotustyökalu antaa suuntaa vetojen suunnitteluun, mutta tämä ei ole lopullinen versio, vaan johdotuksia täytyi muokata manuaalisesti. Lopulliseen version muokkaamiseen meni useita päiviä. Lopulliseen versioon tehtiin painatusmerkinnät, jotka helpottavat käyttöä.

5.2.2 Komponenttien valinta ja hankinta

Hankintadokumentaatiot tehtiin piirilevylle ja tarvittaville komponenteille. Tarjouspyynnöt jätettiin 55 kappaleen määrästä.

PADS-suunnitteluohjelman tuottama komponenttiluettelo (BOM) toimitettiin komponenttimaahantuoille. Komponenttien tarjousten vertailu ja valintaperusteet:

- Sopivuus, vähimmäistilausmäärä.
- Saatavuus, toimitusaika.
- Hinta.

Kyseinen sarjan pienuus ei oikeuttanut suuriin määräalennuksiin, ja osan komponenttien toimitusajat olivat suhteettoman pitkiä. Ennen lopullista komponenttitilausta tehtiin tarjousten perusteella muutoksia piirikaavioon ja korjattiin piirikorttiin tulleet ristiriitaisuudet. Komponentit hankittiin usealta toimittajalta, ja passiiviset komponentit ostettiin ladontayrityksen varastosta.

Piirikorttipohjan valmistaminen vaatii useita erillisiä dokumentteja:

- Johdintasot, tiedostoja 4 kpl

- Juotteenestopinnoite, tiedostoja 2 kpl
- Painatus, tiedostoja 2 kpl
- Poraustiedosto
- Mitoitus, sisältyy usein johdintasokuvauksiin.

Ensimmäistä sarjaa tilattaessa tulee ottaa huomioon muutamia ns. aloituskustannuksia. Piirilevyn toimittajan aloitusmaksu on luokkaa 300 euroa, joka kattaa mm. tiedostojen oikeellisuuden tarkistamisen. Muita aloituskustannuksia ovat kokoonpanossa tarvittavien stensiilien hankinta. Nämä ovat kertaluonteisia kustannuksia, joten seuraavaa sarjaa tilattaessa näitä ei peritä. Korttipohjien ensimmäisen erän hinta oli 3100 euroa ja seuraava tilaus 400 euroa.

Korttipohjat ja stensiilikalvot tilattiin PCB Connect Oy:stä. Piirikortin johdinvahvuudeksi määriteltiin 35 µm ja pinnoitusmateriaaliksi immersio Ni/Au, joka ei hapetu helposti.

5.3 Tuotanto ja testaus

Ennen kokoonpanoa varmistetaan komponenttien kotelotyyppit, että ne vastaavat osaluetteloa. Ohjelmointiyksikön mikrokontrolleri täytyy ohjelmoida ennen kokoonpanoa. Mikrokontrolleri AT-Mega16 on 44-pinnisessä TQFP-kotelossa. Ohjelmointi voidaan tehdä STK600-alustalla tai erillisellä ohjelmointilaitteella, TQFP-44 kotelotyyppille piti hankkia sovitinkortti ohjelmointilaitteeseen. Kokoonpanon jälkeen ohjelmointi on vaikeaa, ja ongelman muodostaa kellon valinta. Tehdasasetuksissa on valittuna sisäinen kello, ja se ei toimi USB/sarjaliikennemuuntimen kanssa. Mikrokontrollerille ladataan ohjelmatiedosto ja päivitetään sisäiset asetukset (fuses) 0xD8, 0xFF.

Hankitut piirikorttipohjat ja komponentit toimitettiin kokoonpanoa varten ladottavaksi.

5.3.1 Piirikorttien ladonta

Komponenttien ladonta tehtiin kajaanilaisessa Elektroval Oy:ssä. Yritys on erikoistunut pienten sarjojen toimittamiseen. Kokoonpanon ensimmäisessä vaiheessa tehdään ladontakoneelle ohjelma, ohjelman tekoon pitää varata noin viikko. Ladontaohjelmiston teossa voidaan hyödyntää

piirilevyohjelmiston tiedostoja. Kortille tehtävät muutokset aiheuttavat ladontakoneen ohjelman päivittämisen ja juotosstensiilien uusimisen, mistä tulee lisäkustannuksia. Ladontatyön kustannus oli n. 2500 euroa ja toimitusaika muutama viikko.

5.3.2 Testaus

Testaaminen suunniteltiin tehtäväksi kymmenen kappaleen sarjoissa. Testausta varten tehtiin testiohjelma, joka testasi kortin kytkimet ja ledit eri porteissa. Testissä SW & LEDS-liittimestä kytkettiin lattakaapeli testattavaan mikrokontrollerin porttiin. Testaus ilman ohjelmaa olisi myös toiminut; debuggaustilassa mikrokontrollerin yksittäistä lähtöä voidaan tarkastella tai sen tilaa voidaan muuttaa ohjelmointikäyttöliittymän avulla.

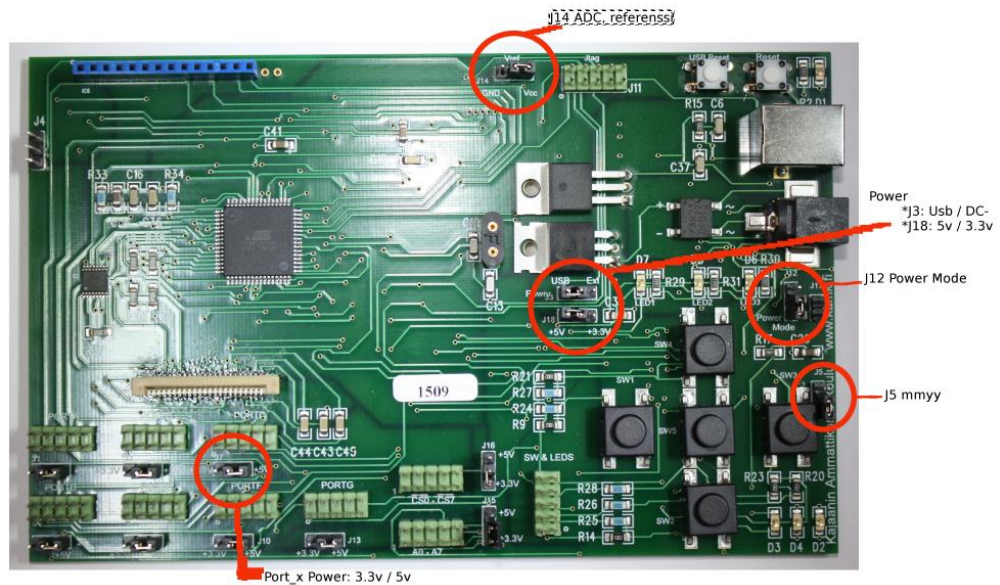
5.3.3 Toiminnallinen testaus

Opetuskorttien testaaminen tehtiin kahdessa osassa. Ensimmäisessä osassa tarkasteltiin visuaalisesti, että kaikki komponentit ovat paikallaan. Komponenttien ladonnan yhteydessä puutteelliset kortit oli merkattu pienillä nuolitarroilla. Samassa yhteydessä kortille laitettiin jumpperointi (kuva 42). Jumpperoinnilla asetetaan kortille esim. jännitesyöttö ja logiikan jännitetaso. Seuraavassa vaiheessa kortille kytkettiin jännitteet. Tässä vaiheessa mitattiin kortin ottama virta, joka pitää olla 40 ... 45 mA. Kehitysalustan virtalähde tuottaa sisäiset käyttöjännitteet 5 ja 3,3 V ja ne mitattiin porttien jännitevalinnan jumpperointiiliittimiltä.

Ohjelmallinen testaus aloitettiin testaamalla AVR-studio-ohjelmistolla Jtag-liittynän kautta yhteys mikrokontrollerille. Mikrokontrollerin sisäiset asetukset testattiin ja asetettiin:

- EXTENDED: 0xFF
- HIGH: 0x19
- LOW: 0xE1

Näillä asetuksilla kortti toimii sisäisellä oskillaattorilla 1 MHz:n taajuudella, mikä mahdollistaa merkkipohjaisen LCD-näytön käyttämisen. Seuraavassa mikrokontrollerille ladattiin testausohjelma.



Kuva 42. Opetuskortin perusjumpperointi

Ohjelmallinen lataus testaa mikrokontrollerin liitännät, testattava portti kytketään lattakaapelilla SW&LEDS-liityntään. Alustetaan testattavan portin suuntarekisteri (DDR_x) debuggaustilassa 0xE0.

Taulukko 5. Kytkimet ja Ledit portissa, alustus 11100000 = 0xE0

<u>PORTx 7</u>	<u>PORTx 6</u>	<u>PORTx 5</u>	<u>PORTx 4</u>	<u>PORTx 3</u>	<u>PORTx 2</u>	<u>PORTx 1</u>	<u>PORTx 0</u>
LED_3	LED_2	LED_1	SW_5	SW_4	SW_3	SW_2	SW_1
1	1	1	0	0	0	0	0

Vialliset ja puutteellisesti ladotut kortit merkattiin ja poistettiin testausalueelta. Testauksen läpäisseet kortit merkittiin värjäämällä sarjanumerotarra vihreällä tussilla.

5.3.4 Ympäristötestaus

Ympäristötestaus tehtiin lähinnä EMC:n osalta. Kajaanin ammattikorkeakoulussa on hyvä testauslaboratorio, jossa tehdään laitetestauksia lähinnä paikallisille yrityksille. Laboratoriota hyödynnetään myös opetuksessa.

5.3.5 Dokumentointi

Kehitysalustaa käytettäessä dokumentoinnin merkitys on suuri, ja varsinkin opetuskäytössä tämä korostuu. Kehitysalustalle tehtiin käyttöönottodokumentti, joka sisältää mm. kortin asetukset (jumpperointi) ja liitindokumentaation. Projektin perustaminen selitetään yksityiskohtaisesti sitä tukevin kuvakaappauksin.

Esimerkkisovellukset helpottavat käyttöönottoa, ja niitä on kehitetty käytön edetessä. Kortin suunnittelutiedostoja voidaan käyttää jatkossa omien sovelluksien kehitystyön pohjana. Kaikki dokumentaatio on opiskelijoiden käytettävissä.

6 Pohdinta

Kehitysalustojen käyttö mahdollistaa sulautettujen järjestelmien kehitystyön, ja niillä on tärkeä rooli myös opetuksessa. Laiteläheistä kehitystyötä tehtäessä kehitysalustat ovat lähes välttämättömyys ja niitä käytetäänkin paljon. Kehitysalustat tukevat niin laitekehitystä kuin ohjelmistokehitystäkin. Kehitysalusta voisi olla jokaisen opiskelijan henkilökohtaisessa käytössä, kortin lisäksi tarvitaan vain USB-kaapeli. Ohjelmistot ovat vapaasti ladattavissa valmistajan nettisivuilta. Käytökynnyks pyritään saamaan mahdollisimman pieneksi.

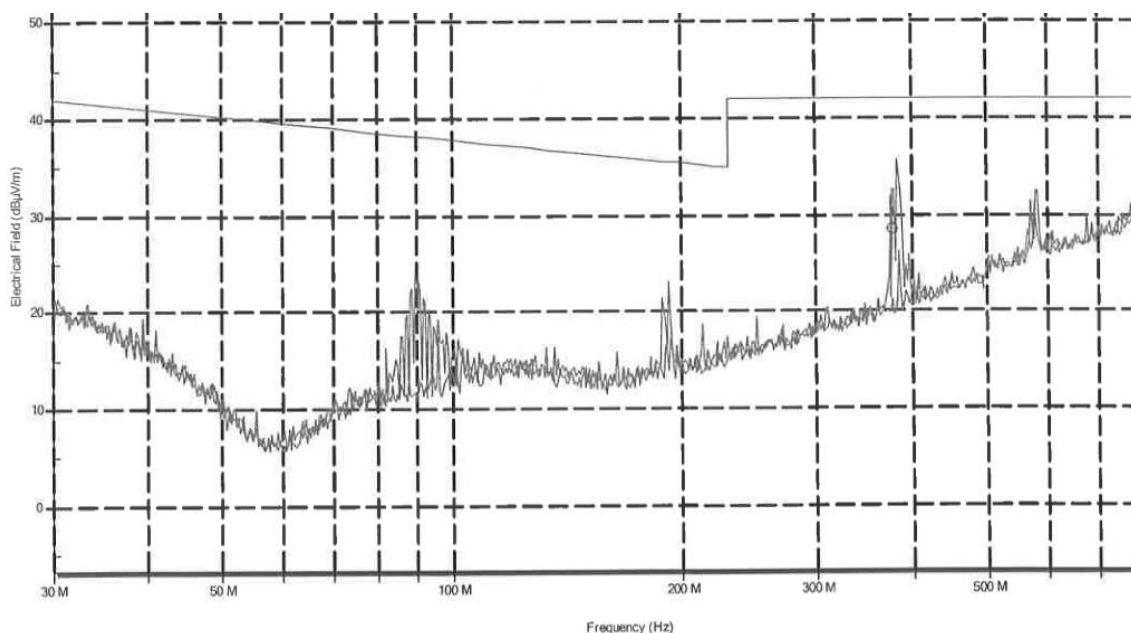
Opetuksessa tulee painottaa kehitysalustojen merkitystä tuotekehitystyökaluna, lopulliseen tuotteeseen ne eivät missään olosuhteissa sovellu. Kehitysalusta ja sen toimintalohkojen kytkennät ovat esimerkkikytkentöjä, joita kopioidaan omiin tuotteisiin. Tämä on usein ongelma käytettäessä käyttöjärjestelmäpohjaisia kehitysalustoja, esim. Raspberry tai Arduino Yun. Näissä tapauksissa piirikortin suunnittelu ja toteutus tulee olemaan haastavaa. Lopullista laitetta suunniteltaessa liityntöjen suojaukset tulevat tärkeään rooliin elektroniikan kestävyuden kannalta, ja näitä suojauksia kehitysalustoissa ei käytetä.

Pelkästään taloudellisesti oman piirikortin kehittämistä ei voida perustella (taulukko 6). Oman kortin käytöstä saatava hyöty on oman tietotaidon nostaminen sekä opetusmateriaalin pitäminen ajan tasalla.

Taulukko 6. Kustannusten vertailu

Kamk-opetuskortti		Kaupallinen
Sisäinen	Jtag-ohjelmointi	Olimex: AVR-JTAG-USB-A. 20€
Sisäinen	I/O- näppi & Led	Sisäinen
Graafinen ja merkkipohjainen	Näyttö	7-seg ja LED 8 kpl
Hinta	50 €	35€
Yhteensä	n. 50 €	n. 55 €

Tässä projektissa on onnistuttu suunnittelemaan yksinkertainen ja toimiva kokonaisuus. Opetuskäytössä kortteja käsitellään ilman ESD-suojauksia, mikä altistaa komponentit sähköstaattisille purkauksille. Tästä huolimatta kortteja on särkynyt melko vähän. Koska kytkentöjä tehdään usein väärin, on kortti suojattu väärin kytkentöjen varalta eikä virtalähdettä saa rikki väärin kytkemällä. Edellä esitettyjen ominaisuuksien myötä kehitysalustat ovat toimineet hyvin.



Kuva 43. EMC-mittausraportti

Opetuskortin EMC-säteilevät häiriöt mitattiin kaiuttomassa huoneessa 30–1000 Mhz. Mittaus on 10 dBμV alle raja-arvon (kuva 43).

Ohjelmointiympäristö on ehkä aikansa elänyt. Uudet versiot vaativat ulkoisen ohjelmointilaitteen hankintaa, joka lisää kustannuksia. Halvimmillaan ulkoinen ohjelmointilaite maksaa muutamia kymmeniä euroja.

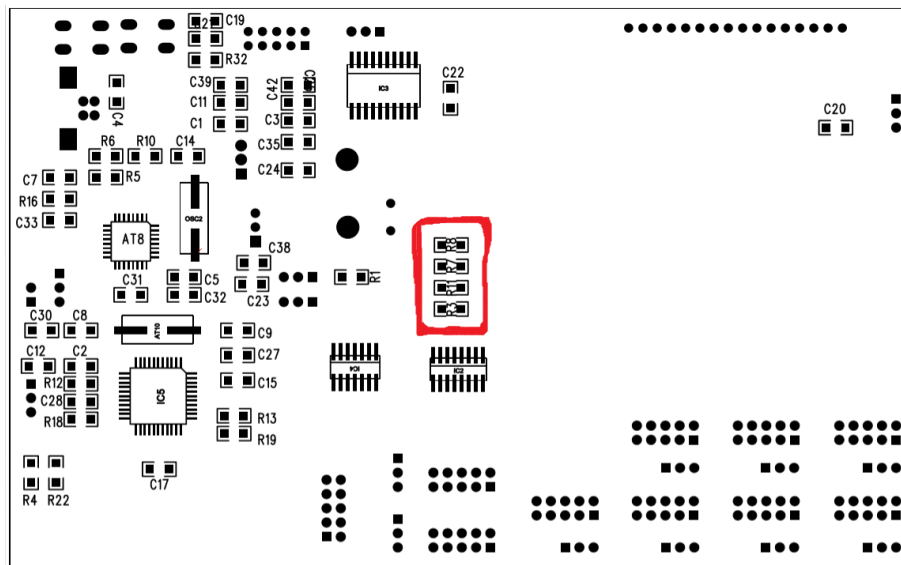
Kortista on tehty muutamia kehitysversioita vuosien varrella.

Rew_2:

Kortilla ollut kiihtyvyyssanturi poistettiin. Anturi oli liitetty kiinteästi SPI-väylään, mikä aiheutti ongelmia B-portin normaaleissa GPIO-toiminnoissa. Lisäksi porttiliitännät keskitettiin yhdelle alu-
eelle ja järjesteltiin selkeästi.

Rew_3:

Kortille lisättiin Jtag-liityntä, joka mahdollistaa ulkoisen ohjelmointilaitteen käytön. Valmistajan tarjoamat uudet IDE-versiot eivät tue vanhaa Jtag ICE -ohjelmointilaiteliityntää. Standardin mukainen Jtag-liityntä on 10-pinninen liitinrima. Liitynnän kautta korttia voidaan ohjelmoida uusilla ohjelmointilaitteilla, jotka toimivat uusien IDE-versioiden kanssa. Omaa tuotetta toteutettaessa liityntä toimii mallikytkentänä, joka voidaan kopioida. Poistamalla vastukset R3, R7, R8 ja R11 katkaistaan datalinjat kortin ohjelmointiyksikköön (kuva 44); tämä mahdollistaa ulkoisen ohjelmointilaitteen käytön.



Kuva 44. Kiinteän ohjelmointiyksikön irrotus

Kortin liitinmerkintöjä parannettiin edelleen.

7 Havaitut kehityskohteet

Kehitysalustaa on käytetty Kajaanin ammattikorkeakoulussa mikrokontrolleritekniikan opetusympäristönä. Kehitysalustat annetaan opiskelijoiden käyttöön valmiiksi alustettuna. Käytännössä alustuksia on kahdenlaisia:

- merkkipohjainen näyttö
- graafinen näyttö.

Asetukset eroavat vain sisäisen kellotaajuden osalta.

Seuraavaan hankittavaan revisioon kannattaa tutkia mahdollisuutta uuden ohjelmointilohkon päivittämiseen.

Komponenttimerkintöjä päivitetään:

- Sijoittelu, LEDs D7, D5, D6. Järjestys sekava.
- Merkinnot liian pienellä.
- Reset-kytkinten merkinnot ristissä.

8 Yhteenveto

Työn aiheena oli kehittää Kajaanin Ammattikorkeakoululle sulautettujen järjestelmien mikrokontrolleriopetukseen uusi kehitysalusta. Aikaisemmin käytössä ollut 8051 - pohjainen opetusympäristö oli vanhentunut ja sen päivittäminen ei ollut järkevää.

Aluksi vertailtiin muissa oppilaitoksissa kehitettyjä kehitysalustoja sekä erilaisia kaupallisia toteutuksia. Kehitysalustan tuli olla helppokäyttöinen ja luotettava, myös käyttöönotto helppoa ilman ylimääräisiä ulkoisia laitteita. Vertailujen perusteella päädyttiin kehittämään oma sulautettujen järjestelmien opetusympäristö.

Mikrokontrolleriksi valikoitui yleiskäyttöinen Microchipin ATmega 128, joka on kestänyt hyvin aikaa ja siinä on riittävästi ominaisuuksia opetuskäyttöön. Piirin valmistajan tarjoamat ohjelmointityökalut ovat vapaasti ladattavissa mikä osaltaan madaltaa käyttöönotto kynnystä. Piiristä on tarjolla hyvä dokumentaatio, ja se sisältää kattavan valikoiman ominaisuuksia opetuskäyttöön. Kehitysalustalle toteutettiin JTAG-ohjelmointilaite, mikä mahdollistaa mikrokontrollerin ohjelmoinnin ja ohjelman testaamisen käsky kerrallaan. Kehitysalusta suunniteltiin 4-kerros piirikortille, tämä mahdollistaa hyvän jännite syötön komponenteille ja estää sähköistenhäiriöiden etenemisen kortilla. Ulkoinen jännitesyöttö suojattiin väärin kytkennöiltä ja kehitysalustalla on vaihdettavat logiikka jännitetasot.

Opetuskortti on toiminut opetuskäytössä hyvin ja siinä on riittävästi ominaisuuksia. Kortille on tehty joitain parannuksia vuosien saatossa mutta perusratkaisut ovat säilyneet. Liityntöjen merkitseminen on opetuskäytössä erittäin tärkeää niin onkin parannettu jokaisessa kehitysversiossa.

Tulevaisuudessa kehitysympäristöä voi kehittää niin, että uusimpien ohjelmistotyökalujen käyttö olisi mahdollista. Tämä vaatii ohjelmointiyksikön päivittämisen uudempaan versioon. Tänä päivänä sulautettuja järjestelmiä kutsutaan IoT-järjestelmiksi. Jos opetuskorttia käytetään osana IoT-järjestelmää, korttiin tulisi suunnitella rajapinta langatonta tiedonsiirtoa varten. Mikrokontrollerista on olemassa versio (ATmega128RFA1), joka tukee 2,4 GHz:n tiedonsiirtoa. Tämä mahdollistaa tiedonsiirron ZigBee- ja IEEE 802.15.2 -standardin mukaisesti.

Lähteet

Kirja:

1. AVR Programming: Elliot Williams (2014). Maker Media

Internet-lähde (ei tietokannoista):

2. Sähköalan standardi kokoelmat, käsikirjat ja direktiivit erisovellus aloittain, <https://www.sesko.fi/standardit>. Haettu 08.02.2020.
3. Silicon labs kokeilukortin esittelysivut biometric sensor and Bluetooth, <https://www.silabs.com/products/development-tools/sensors/environmental-biometric-sensor-puck-starter-kit>. Haettu 01.02.2020.
4. STM32 Nucleo Boards, valmistajan kotisivut, <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>. Haettu 20.1.2020.
5. Raspberry kehitysalustan käyttöjärjestelmät, <https://www.raspberrypi.org/downloads/>. Haettu 03.03.2020
6. Microchip Ltd. mikrokontrollerivalmistajan kotisivut, www.Microchip.com. Haettu 12.12.2019
7. Eletroniikka komponenttien hintoja, suurten toimittajien internetsivuilta, esim. www.farnell.com www.Mouser.com
8. Opetusvideo Tina TI -simulaatio-ohjelman käytöstä, <https://www.youtube.com/watch?v=IG2WcFDitzE> Haettu 1.3.2020.
9. Siemensn pads-elektroniikkasuunnittelutyökalut, www.Mentor.com. Haettu 2.3.2020.
10. Autodeskin opiskeluun tarjoamat ilmaisohjelmat, <https://www.autodesk.com/education/free-software/featured>. Haettu 2.3.2020.

11. Jtag-liittymän standardi, https://standards.ieee.org/standard/1149_1-2013.html. Haettu 4.3.2020.
12. Tukes CE-merkintä, <https://tukes.fi/tuotteet-ja-palvelut/ce-merkinta>. Haettu 5.3.2020.
13. HS-Heilbronn UAS, <https://www.hs-heilbronn.de/en>. Haettu 5.3.2020.
14. Seinäjoen AMK Surfnet. Tapio Varmola, suomalaisen ammattikorkeakoulun rakentaja ja kehittäjä. Ympäristö 3.1 kehitysympäristöt s. 290. Haettu 5.3.2020.
15. VTI:n CMR3000-kiihtyvyyssanturi, <https://media.digi-key.com/pdf/Data%20Sheets/VTI%20Technologies%20PDFs/CMR3000-D01.pdf>. Haettu 8.3.2020.
16. ce-merkintä. Tukes tuotteet ja palvelut. haettu internetosoite <https://tukes.fi/tuotteet-ja-palvelut/ce-merkinta>
17. Alkuperäinen AVR-Jtag ohjelmointilaite. Aquaticus ROV team, 2006 Haettu osoitteesta: <http://Aquaticus.info/jtag>

Android play kauppa:

AVR Tutorial. (4.3.2020) PeterHo

C Programming Tutorials (15.2.2020)

Internet-video (esimerkiksi YouTube):

AVR Programming- AVR Studio Tutorial Introduction, haettu osoitteesta <https://www.youtube.com/watch?v=plJf0r7lcWc>

AVR – IoT WG, haettu osoitteesta <https://www.youtube.com/watch?v=WK4llyKDMIQ>

